AD-A146 606

DTIC FILE COPY

September 1984 LIDS-P-1404

A PERSPECTIVE ON MULTIACCESS CHANNELS*

by

R. G. Gallager

ABSTRACT

The Information Theoretic Approach and the Collision Resolution Approach to Multiaccess Channels are reviewed in terms of the Underlying Communication Problems that both are modelling. The author gives some perspective on the strengths and weakness of these approaches and argues for the need of a more combined approach focused on coding and decoding techniques.

DTIC ELECTE OCT 16 1984 A

*This research was conducted at the M.I.T. Laboratory for Information and Decision Systems with partial support provided by NSF under Grant NSF-ECS-8310698 and by DARPA under Contract ONR/N00014-84-K-0357.

84 10 09 081

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A146604
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) A PERSPECTIVE ON MULTIACCESS CHANNELS	5. TYPE OF REPORT & PERIOD COVERED Technical
	6. PERFORMING ORG. REPORT NUMBER LIDS-P-1404
7. AUTHOR(s) Robert G. Gallager	8. CONTRACT OR GRANT NUMBER(s) DARPA Order No. 3045/2-2-84 Amendment #11 ONR/N00014-84-K-0357
9. PERFORMING ORGANIZATION NAME AND ADDRESS Massachusetts Institute of Technology Laboratory for Information & Decision Systems Cambridge, Massachusetts 02139	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Code No. 5T10 ONR Identifying No. 049-383
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agence 1400 Wilson Boulevard Arlington, Virginia 22209	12. REPORT DATE September 1984
	13. NUMBER OF PAGES 78
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research Information Systems Program Code 437 Arlington, Virginia 22217	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

DTIC COPY INSPECTED

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Accession For
NTIS GRA&I
DTIC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Information Theoretic Approach and the Collision Resolution Approach to Multiaccess Channels are reviewed in terms of the Underlying Communication Problems that both are mdelling. We give some perspective on the strengths and weakness of these approaches and argue for the need of a more combined approach focused on coding and decoding techniques.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

A PERSPECTIVE ON MULTIACCESS CHANNELS

I. INTRODUCTION

This paper is an expanded version of the Shannon lecture at the International Symposium on Information Theory at St. Jovite, Quebec, in September, 1983. For the last ten years there have been at least three bodies of research on multiaccess channels, each proceeding in virtual isolation from the others and each using totally different models. The objective here is to contrast these bodies of work and to give some perspective on what is needed to provide some unification between the areas. We shall refer to the three areas as collision resolution, multiaccess information theory, and spread spectrum.

The kind of communication situation that these three areas address is illustrated in fig. 1.1. There are multiple transmitters and a single receiver. The received signal is corrupted both by noise and by mutual interference between the transmitters. Each of the transmitters is fed by an information source, and each information source generates a sequence of messages, successive messages arriving at random instants of time. There is usually some small amount of feedback from the receiver to the tranmitter, but this feedback will not be our main focus. Our major focus, rather, is on the interference, the noise, and the random, or "bursty", message arrivals.

This type of model is appropriate for the up link of a satellite network, for a radio network where there is one central repeater, and for the traffic to the central node on a multidrop

telephone line. It is also adequate in most respects for studying networks where a common channel allows all nodes to hear all other nodes. Common examples are a cable connecting many nodes and a fully connected radio network.

The beginning of the collision resolution approach to multiaccess communication came in 1973 with Abramson's Aloha network [1]. The idea here was that whenever a message (or packet) arrived at a transmitter, it would simply be transmitted, ignoring all other transmitters in the network. If another transmitter was transmitting in an overlapping interval, interference would prevent the message from being correctly received, the cyclic redundancy check (CRC) would not check, no acknowledgement would be sent, and the transmitter would try again later; the later time would be pseudorandomly chosen to avoid the certainty of another collision if both transmitters waited the same time.

Over the years, this basic strategy has been improved, generalized, and analyzed in many ways. A number of variations are in widespread use, and the general topic of collision resolution has provided many challenging and interesting problems for research. Section 4 provides an introduction to these problems and most of the other papers in this special issue are devoted to the current state of these problems.

Collision resolution research has always focused on the bursty arrivals of messages and the interference between transmitters, but has generally ignored the noise. More generally, this approach ignores the underlying communication

process, assuming only that a message transmission is correctly received in the absence of collision and incorrectly received otherwise.

The multiaccess information theoretic approach to multiaccess also started in 1973 with a coding theorem developed by Ahlswede [2] and Liao [3]. This work has also been generalized in many ways and has opened up a separate area of research problems. Excellent summaries and descriptions of this research are given in [4,5,6]. In this approach, the noise and interference aspects of the multiaccess channel are appropriately modelled, but the random arrivals of the messages are ignored.

Before proceeding, it is important to understand why information theorists and communication system designers have always essentially ignored random message arrivals for point to point channels, and why this is usually unreasonable for multiaccess channels. For a point to point channel, one normally assumes an infinite reservoir of data to be transmitted. The reason for this is that it is a minor practical detail to inform the receiver when there is no data to send; furthermore there is no other use for the channel, so potential lack of data might as well be left out of the model. For multiaccess channels, on the other hand, most transmitters have nothing to send most of the time, and only a few are busy. The problem is then to share the channel between the busy users, and this is often the central technical problem in multiaccess communication.

A pure theoretician would properly point out here that bursty message arrivals have nothing to do with coding theorems

for multiaccess channels. The arrivals have to do with the sources and can and should be dealt with through source coding. Even without source coding, if the arrival process is ergodic, then over the arbitrarily long time intervals used in the coding theorems, the bursty arrivals will not matter.

From a more practical point of view, the limit theorems of information theory are interesting both because they put an upper limit on what is achievable and because the limit is usually not too far from what is practically achievable. For a multiaccess channel, however, the long time intervals required for the source arrivals to appear smoothed out are typically far greater than the tolerable delays. Conversely, the time interval required for coding to be effective (ie. the time for the noise to be smoothed out) is typically smaller than the tolerable delay. What is needed then is an information theoretic model that somehow precludes the possibility of imposing long delays on source messages.

One approach to this, which is used in the collision resolution field, is to assume an infinite number of sources, or equivalently, that a new transmitter is created for each new arriving message and then destroyed when the message is successfully transmitted. The received sequence or waveform would then be some function of noise and whatever was being transmitted by the active transmitters. It seems that to develop understanding in this area, it is necessary first to develop some understanding of coding (as opposed to coding theorems) in a multiaccess environment. This understanding should involve

decoding in the presence of several messages being transmitted simultaneously, since otherwise the problem simply reduces to conflict resolution with coding added for reliable transmission in the absence of conflicts.

In section 2, we discuss multiaccess information theory in more detail, and in section 3, we discuss what little is known about coding. In both sections, the discussion is restricted to systems with only two sources. It appears to be important to understand coding in this simplest context before tackling the problem of real interest with many sources and transmitters.

The spread spectrum approach to multiaccess channels [7,8] will not be discussed in any detail in this paper, but is briefly discussed here in order to illustrate the types of possibilities for multiaccess communication that lie outside the conventional collision resolution and coding theory approaches. Spread spectrum is a mode of communication originally developed to protect against jamming in a military environment. The signal to be transmitted is modulated over a much broader frequency band, say β times more than necessary. Assuming that the jammer does not know the modulating sequence, the jammer's signal will essentially look like broad band noise to the signal, and the noise seen by the receiver after demodulation will be reduced by a factor of β.

For multiaccess communication using spread spectrum, several sources can transmit at once using different modulating sequences, and each will look like broad band noise to the others. If we compare this type of system to frequency

multiplexing, using β frequency bands, it appears at first that spread spectrum is not a very good idea. When a number of transmitters approaching β transmit together using spread spectrum, the self noise becomes considerable, and the resulting system is clearly inferior to FDM in terms of capacity. The problem with FDM, however, is that if there are many more than β transmitters in the system, but typically many fewer than β with messages to send, there is a problem allocating the frequencies to the busy transmitters (this is the same fundamental problem handled by the collision resolution approach). Since many times more than β modulation sequences can be chosen that are almost orthogonal and look like noise to each other, spread spectrum provides an automatic solution to the problem of allocating the channel to the busy users. This solution is not entirely satisfactory, since one still needs collision resolution when too many transmitters send at once, and the decoding is very complex. It illustrates, however, a major point of this paper - namely that a more fundamental approach and set of models are needed for multiaccess communication than the collision resolution or information theoretic approaches alone.

II. The Information Theoretic Approach

The coding theorems of information theory treat the question of how much data can be reliably communicated from one point, or set of points, to another point, or set of points. It is tacitly assumed that the sources have a never emptying reservoir of data to send. Thus the theoretical results in this area do not address the question of the delay that arises in multiple access systems because of the random arrival times of data to be transmitted.

The class of channels to be considered is illustrated in Fig. 2.1. Each unit of time, the first transmitter sends a symbol x from an alphabet X and the second transmitter sends a symbol w from an alphabet W. There is an output alphabet Y and a transmitter probability assignment $P(y|xw)$ determining the probability of receiving each $y \varepsilon Y$ for each choice of inputs $x \varepsilon X$, and $w \varepsilon W$. The channel is memoryless in the sense that if $\mathbf{x} = (x_1,\ldots,x_N)$ and $\mathbf{w} = (w_1,\ldots,w_N)$ represent the inputs to transmitters one and two respectively over N successive time units, then the probability of receiving $\mathbf{y} = (y_1,\ldots,y_N)$ for the given $\mathbf{x},\mathbf{w}$, is

$$P(\mathbf{y}|\mathbf{xw}) = \prod_{n=1}^{N} P(y_n|x_n w_n) \qquad (2.1)$$

We assume for the time being that the alphabets are all discrete, but it will soon be obvious that this can be generalized in the same way as for single input channels.

As indicated in the figure, there are two independent

sources which are encoded independently into the two channel inputs. Consider block coding with a given block length N and with M code words, $(x_1, x_2, \ldots, x_M)$, for transmitter 1 and L code words $(w_1, \ldots, w_L)$ for transmitter 2; each code word is a sequence of N channel inputs. For convenience we refer to a code with these parameters as an (N,M,L) code. The rates of the two sources are defined as

$$R_1 = (\ln M)/N, \quad R_2 = (\ln L)/N \tag{2.2}$$

Each N units of time, source 1 generates an integer m uniformly distributed from 1 to M and source 2 independently generates an integer ℓ uniformly distributed from 1 to L. The transmitters send x_m and w_ℓ respectively, and the corresponding channel output y enters the decoder and is mapped into a decoded "message" $\hat{m}, \hat{\ell}$. If both $\hat{m} = m$ and $\hat{\ell} = \ell$, the decoding is correct and otherwise a <u>decoding error</u> occurs. The probability of decoding error, P_e is minimized for each y by a maximum likelihood decoder, choosing $(\hat{m}, \hat{\ell})$ as integers $1 \le m' \le M$, $1 \le \ell' \le L$ that maximize $P(y|x_{m'}, w_{\ell'})$. If the maximum is non-unique, any maximizing (m', ℓ') can be chosen with no effect on P_e. Both sets of code words $(x_1, \ldots, x_M)$ and $(w_1, \ldots, w_L)$ are known to the decoder, but, of course, the source outputs m, ℓ are unknown.

The most fundamental result about these channels is the coding theorem due to Ahlswede [2] and Liao [3]. Let $Q_1(x)$ and $Q_2(w)$ be probability assignments on the X and W input alphabets respectively. Define the achievable rate region R as the convex hull of the set of rate pairs (R_1, R_2) which, for some choice of

assignments Q_1, Q_2, satisfy each of the inequalities:

$$R_1 + R_2 \leq I(XW;Y) = \sum_{x,w,y} Q_1(x)Q_2(w)P(y|xw)\ln \frac{P(y|xw)}{P(y)} \qquad (2.3)$$

$$0 \leq R_1 \leq I(X;Y|W) = \sum_{x,w,y} Q_1(x)Q_2(w)P(y|xw)\ln \frac{P(y|xw)}{P(y|w)} \qquad (2.4)$$

$$0 \leq R_2 \leq I(W;Y|X) = \sum_{x,w,y} Q_1(x)Q_2(w)P(y|xw)\ln \frac{P(y|xw)}{P(y|x)} \qquad (2.5)$$

where $P(y) = \Sigma_{xw}\, Q_1(x)Q_2(w)P(y|xw)$, $P(y|w) = \Sigma_x\, Q_1(x)P(y|xw)$, and $P(y|x) = \Sigma_w\, Q_2(w)P(y|xw)$.

The region bounded by (2.3)-(2.5) for a given Q_1, Q_2 is shown in fig. 2.2. It is easy to see that the break points of the boundary occur at $R_1 = I(X;Y|W)$, $R_2 = I(W;Y)$ and at $R_1 = I(X;Y)$, $R_2 = I(W;Y|X)$. In general $I(X;Y|W) \geq I(X;Y)$ with equality iff x and w are conditionally independent given y.

Theorem 2.1 (Ahlswede,Liao): For each $\varepsilon > 0$, $\delta > 0$, $(R_1,R_2) \in \mathbf{R}$, there exists an N_o such that for all $N \geq N_o$, $M \leq \exp N(R_1-\delta)$, $L \leq \exp N(R_2-\delta)$, there exists an (N,M,L) code with $P_e \leq \varepsilon$. For each $\delta > 0$ and $(R_1,R_2) \in \mathbf{R}$, there exists $\varepsilon > 0$ such that $P_e \geq \varepsilon$ for all (N,M,L) codes with $M \geq \exp N(R_1+\delta)$, $L \geq \exp N(R_2+\delta)$.

In effect, the theorem says that reliable communication is possible for source rates in the interior of the achievable

region and is impossible outside of the achievable region. Slepian and Wolf [9] later generalized this result by considering a third source that could be encoded jointly for both transmitters. They also used a random coding argument which showed both that P_e can be made to decrease exponentially with N and also, in a sense, that most codes have this behavior. Since this random coding argument is a very simple extension of random coding for single input channels and it gives a great deal of insight into coding for multiple access channels, we now go through the argument for the two source case.

Let $Q_1(x)$ and $Q_2(w)$ be probability assignments on the X and W alphabets respectively and consider an ensemble of (N,M,L) codes where each code word x_m, $1 \le m \le M$ is independently selected according to the probability assignment

$$Q_1(\mathbf{x}) = \prod_{n=1}^{N} Q_1(x_n), \quad \mathbf{x} = (x_1, x_2, \ldots, x_N) \tag{2.6}$$

and each code word w_ℓ, $1 \le \ell \le L$ is independently selected according to

$$Q_2(\mathbf{w}) = \prod_{n=1}^{N} Q_2(w_n), \quad \mathbf{w} = (w_1, \ldots, w_N) \tag{2.7}$$

For each code in the ensemble, the decoder uses maximum likelihood decoding, and we want to upper bound the expected value $\bar{P}_e$ of P_e for this ensemble. Define an error event to be of type 1 if the decoded pair $(\hat{m},\hat{\ell})$ and the original source pair

(m,ℓ) satisfy $\hat{m} \neq m$, $\hat{\ell} = \ell$. An error event is type 2 if $\hat{m} = m$ and $\hat{\ell} \neq \ell$, and is of type 3 if $\hat{m} \neq m$ and $\hat{\ell} \neq \ell$. Let P_{ei}, $1 \leq i \leq 3$, be the probability, over the ensemble, of a type i error event; obviously $P_e = P_{e1} + P_{e2} + P_{e3}$.

Consider P_{e3} first. Note that when (m,ℓ) enters the encoder, there are M-1 choices for $\hat{m}$ and (L-1) choices for $\hat{\ell}$, or (M-1)(L-1) pairs, that yield a type 3 error. For each such pair $(\hat{m},\hat{\ell})$, the code word pair $\mathbf{x}_{\hat{m}}$, $\mathbf{w}_{\hat{\ell}}$ is statistically independent of $\mathbf{x}_m$, $\mathbf{w}_\ell$ over the ensemble of codes. Thus, regarding $(\mathbf{x},\mathbf{w})$ as a combined input to a single input channel with input alphabet XxW, we can directly apply the coding theorem, theorem 5.6.1 of [10], which asserts* that for all ρ, $0 \leq \rho \leq 1$,

$$P_{e3} \leq [(M-1)(L-1)]^{\rho} \sum_{\mathbf{y}} \left[\sum_{\mathbf{x},\mathbf{w}} Q_1(\mathbf{x}) Q_2(\mathbf{w}) P(\mathbf{y}|\mathbf{x}\mathbf{w})^{1/(1+\rho)} \right]^{1+\rho} \qquad (2.8)$$

*The statement of thoerem 5.6.1 of [10] assumes that all code words are chosen independently, but the proof only uses pairwise independence between the transmitted word ($(\mathbf{x}_m,\mathbf{w})$ in the case here) and each other word ($(\mathbf{x}_{\hat{m}},\mathbf{w}_{\hat{\ell}})$ $\hat{m} \neq m$, $\hat{\ell} \neq \ell$ for the case here).

Using the product form of Q_1, Q_2, and P, Eqs. (2.1, 2.6, 2.7), and the definition of rates in (2.2), this simplifies to

$$P_{e3} \leq \exp[\rho N(R_1+R_2)] \left[\sum_y \left[\sum_{x,w} Q_1(x)Q_2(w)P(y|xw)^{1/(1+\rho)}\right]^{1+\rho}\right]^N \quad (2.9)$$

Next consider P_{e1}, the probability that $\hat{m} \neq m$ and $\ell = \hat{\ell}$. We first condition this probability on a particular message ℓ entering the second encoder, and a choice of code with a particular w_ℓ transmitted at the second input. Given w_ℓ, we can view the channel as a single input channel with input x_m and with transition probabilities $P(y|x_m w_\ell)$.

A maximum likelihood decoder for that single input channel will make an error (or be ambiguous) if

$$P(y|x_m, w_\ell) \geq P(y|x_{m'} w_\ell) \text{ for at least one } m' \neq m. \quad (2.10)$$

Since this event must occur whenever a type 1 error occurs, the probability of a type 1 error, conditional on w_ℓ is upperbounded by the probability of error or ambiguity on the above single input channel. Using theorem 5.6.1 of [10] again for this single input channel, we have, for any ρ, $0 \leq \rho \leq 1$,

$$P[\text{Type 1 error}|w_\ell] \le (M-1)^\rho \sum_y \left[\sum_x Q_1(x)P(y|xw)^{1/(1+\rho)}\right]^{1+\rho} \qquad (2.11)$$

Taking the expected value of (2.11) over w_ℓ and then using the product form of Q_1, Q_2 and P again,

$$P_{e1} \le \exp[\rho N R_1]\left[\sum_{y,w} Q_2(w)\left[\sum_x Q_1(x)P(y|xw)^{1/(1+\rho)}\right]^{1+\rho}\right]^N \qquad (2.12)$$

Applying the same argument to type 2 errors, for all ρ, $0 \le \rho \le 1$,

$$P_{e2} \le \exp[\rho N R_2]\left[\sum_{y,x} Q_1(x)\left[\sum_w Q_2(w)P(y|xw)^{1/(1+\rho)}\right]^{1+\rho}\right]^N \qquad (2.13)$$

Putting (2.9), (2.12), (2.13) in a form to emphasize the exponential dependence on N, we have:

<u>Theorem</u> 2.2 (Slepian-Wolf): Consider an ensemble of (N,M,L) codes in which $(x_1,\ldots,x_m)$ and $(w_1,\ldots,w_L)$ are independently chosen according to (2.6) and (2.7) for a given probability assignment $Q(xw) = Q_1(x)Q_2(w)$. Then the expected error probability over the ensemble satisfies

$$P_e \le P_{e1} + P_{e2} + P_{e3} \qquad (2.14)$$

$$P_{ei} \leq \exp\left[-N[-\rho R_i + E_{0i}(\rho,Q)]\right] \quad \text{for all } \rho,\ 0 \leq \rho \leq 1,\ \text{all } i = 1,2,3. \tag{2.15}$$

$$R_1 = \frac{\ln M}{N}, \qquad R_2 = \frac{\ln L}{N}, \qquad R_3 = R_1 + R_2 \tag{2.16}$$

$$E_{o1}(\rho,Q) = -\ln \sum_{y,w} Q_2(w)\left[\sum_x Q_1(x)P(y|xw)^{1/(1+\rho)}\right]^{1+\rho} \tag{2.17}$$

$$E_{o2}(\rho,Q) = -\ln \sum_{y,x} Q_1(x)\left[\sum_w Q_2(w)P(y|xw)^{1/(1+\rho)}\right]^{1+\rho} \tag{2.18}$$

$$E_{o3}(\rho,Q) = -\ln \sum_y \left[\sum_{x,w} Q_1(x)Q_2(w)P(y|xw)^{1/(1+\rho)}\right]^{1+\rho} \tag{2.19}$$

The behavior of the expressions $E_{oi}(\rho,Q)$, $i = 1,2,3$, is the same as for the single input case. In particular let I_i, $i = 1,2,3$, be given by

$$I_1 = I(X;Y|W), \quad I_2 = I(W;Y|X), \quad I_3 = I(XW;Y) \tag{2.20}$$

as defined in (2.3)-(2.5). Then if $I_i > 0$, the function $E_{oi}(\rho,Q)$ is convex ∩, strictly increasing in ρ, and positive for $\rho > 0$. Furthermore, the maximum of $E_{oi}(\rho,Q)-\rho R_i$ over $0 \leq \rho \leq 1$ is positive and decreasing in R_i for $0 \leq R_i < I_i$ (see theorem

and 5.6.4 of [10] for proofs). Theorem 2.2 then asserts that if $R_i < I_i$, $i = 1,2,3$, then P_e decreases exponentially with increasing N.

There are two questions we want to explore in the rest of this section. First, how tight is this bound on error probability, and second, what indication does it give of the practicality of coding for multiple access channels. To explore the question of tightness, we first interpret the terms P_{ei} in (2.14).

P_{e1}, as upper bounded in (2.12), is the error probability that would result if a "genie" informed the decoder about the second source message ℓ. This genie aided error probability is also clearly a lower bound to P_e, so that when type 1 errors are the predominant cause of errors, the genie aided error probability closely approximates P_e. Similarly, the bound for P_{e3} is the conventional single input random coding bound for a single code of rate R_1+R_2 using combined inputs with probability $Q_1(x)Q_2(w)$. Our conclusion, then, is that the bound on P_e in theorem 5.2 is quite tight for the given ensemble of codes. The problem, as we shall soon see through a set of examples, is that the best codes are not always representative of the ensembles.

Example 1: The Collision Channel.

Let $X = (0,1,\ldots K)$ and $W = (0,1,\ldots,K)$. We regard 0 as an "idle" input, and if 0 is the x input for a given w input, then y is the pair (0,w). Similarly if w=0, the output is (x,0). Finally if $x \neq 0$ and $w \neq 0$, the output y is a special symbol e representing "collision". This is shown in fig. 2.3 for K=2.

First consider the achievable rate region. For any given $Q_1(x), Q_2(w)$, it is easy to see that, conditional on the output y, the two inputs are statistically independent; thus $I(X;Y|W) = I(X;Y)$ and the set of rates satisfying (2.3)-(2.5) forms a rectangle. We next want to find the set of rates so that (2.3)-(2.5) is satisfied for <u>some</u> choice of Q_1, Q_2. It should be clear from symmetry that $Q_1(x)$ should be constant for all $x > 0$ and $Q_2(w)$ should be constant for $w > 0$; thus we need only consider the union of rates satisfying (2.3)-(2.5) over all choices of $Q_1(0)$ and $Q_2(0)$. Fig. 2.4 shows the resulting union; for all $K \geq 8$, the set of rates is non-convex (the potential non-convexity for multi-access channels was first shown by [11]). The convex hull of this union region is the set of achievable rates of theorem 2.1. Theorem 2.2 assures us that exponentially decaying error rates are achievable in the interior of the union region. Any given rate pair in the interior of the convex hull is on a straight line between two pairs of rates each in the interior of the union region. By time division multiplexing between codes for these rate pairs, reliable communication is achieved for the given rate pair. Thus theorem 2.2 establishes the positive half of theorem 2.1.

It is rather surprising at first that the union region is non-convex. We note that $I(XW;Y)$ is a convex $\cap$ function of $Q_1(x)$ and a convex $\cap$ function of $Q_2(x)$, but is non-convex as a joint function of Q_1 and Q_2. It is also convex as a function of $Q(x,w)$, but the set of probability vectors $Q(x,w)$ for which $Q(x,w) = Q_1(x)Q_2(w)$ for some Q_1, Q_2 is a non convex region. Thus

$$E_{oi}(\rho,\lambda,Q^{(1)},Q^{(2)}) = \lambda E_{oi}(\rho,Q^{(1)}) + (1-\lambda)E_{oi}(\rho,Q^{(2)}) \qquad (2.25)$$

No examples have been found where this approach enlarges the regions R_α defined above; this approach is sufficient, however, to achieve exponential decays in P_e for all rate pairs in the interior of **R**.

Another approach is to consider random coding ensembles in which successive letters are statistically dependent. For the collision channel, for example, suppose the block is divided into sub-blocks of four letters each. Within each sub-block, (x_1,x_2,x_3,x_4) has either the form $(x,x,0,0)$ or the form $(0,0,x,x)$, each with equal probability. Similarly, (w_1,w_2,w_3,w_4) has either the form $(w,0,w,0)$ or $(0,w,0,w)$ with equal probability. Finally, x and w are independently and equiprobably chosen from $(1,2,\ldots,K)$. With this arrangement, each sub-block is equivalent to a noiseless x channel with 2K inputs and a noiseless w channel with 2K inputs (this example was suggested by Massey's coding scheme for unsynchronized collision channels [14]). The resulting random coding exponent is clearly larger than that where the successive letters are independent with the same marginal probabilities.

The purpose of the above discussion was not to find the largest exponents achievable for the collision channel, but rather to illustrate why error exponents are far more complicated for multiaccess channels than for single input channels. It also illustrates why there is no simple sphere packing lower bound to P_e for multiaccess channels that yields the same error exponents

as the random coding bound. Arutyunyan [15] has developed a type of sphere packing bound for multiaccess channels, but it is somewhat loose since it does not account for the separation of the two encoders for the type 3 errors.

Example 2: Additive White Gaussian Noise Channel (AWGN)

We now turn to another example of somewhat greater practical importance where the random coding exponents work out more nicely. Suppose the X, W, and Y alphabets are each the set of real numbers, and the output y is given by

$$y = x + w + z \tag{2.26}$$

where z is a zero mean Gaussian random variable of variance σ^2 independent of x and w. The x input and w input are each constrained to have mean square values at most S_1 and S_2 respectively. If we consider the channel as a cascade of a noiseless channel adding x and w and then a single input Gaussian channel, we see that I(XW;Y) is at most the capacity of the single input channel with the input constrained to energy S_1+S_2. Thus

$$I(XW;Y) \leq \frac{1}{2} \log\left[1 + \frac{S_1 + S_2}{\sigma^2}\right] \tag{2.27}$$

It is also easy to see that I(X;Y|W) is the average mutual information between x and y in the absence of w. Thus

$$I(X;Y|W) \leq \frac{1}{2} \log\left[1 + \frac{S_1}{\sigma^2}\right] \tag{2.28}$$

$$I(W;Y|X) \leq \frac{1}{2} \log\left[1 + \frac{S_1}{\sigma^2}\right] \tag{2.29}$$

These inequalities are satisfied for all independent distributions on x and w and are all satisfied with equality if x and w are independent zero mean Gaussian with variances S_1 and S_2 respectively. Thus the rate region for which (2.3)-(2.5) are satisfied for some independent x and w distribution is

$$R_1 + R_2 \leq \frac{1}{2} \log\left[1 + \frac{S_1+S_2}{\sigma^2}\right] \tag{2.30}$$

$$0 \leq R_1 \leq \frac{1}{2} \log\left[1 + \frac{S_1}{\sigma^2}\right] \tag{2.31}$$

$$0 \leq R_2 \leq \frac{1}{2} \log\left[1 + \frac{S_2}{\sigma^2}\right] \tag{2.32}$$

Since this region is convex already, it is the achievable rate region **R**.

This region **R** is sketched in fig. 2.5 for various values of signal to noise ratios $A = S/\sigma^2$, $S = S_1+S_2$, for the case where $S_1 = S_2$. Note that the region is almost rectangular for small A and almost triangular for large A. Note that if one uses TDM between a code for x and a code for w, then the achievable rates are limited to the region bounded by the straight line between

the axis intercepts of the boundary of **R** (see fig. 2.5). Thus for large A, TDM is almost as good as the best coding, whereas for small A, TDM is quite inferior. The reason for this can be seen most clearly for the case $R_1 = R_2$. Alternating between $(R_1,0)$ and $(0,R_2)$ then wastes half the available power, since (by our model), the first transmitter stays within its power limitation while transmitting. Losing half the available power loses only a small fraction of the available capacity for large A whereas, for small A, a large fraction is lost. This suggests using frequency division multiplexing, achieving the same simplicity as TDM, but being able to use all the available power (see fig. 2.6).

Next consider the random coding exponent for these channels. Using the above Gaussian distribution for x and w, we can easily calculate $E_{oi}(\rho,Q)$ from (2.17)-(2.19), replacing sums with integrals. The result is

$$E_{oi}(\rho,Q) = \frac{\rho}{2} \ln\left[1 + \frac{S_i}{\sigma^2(1+\rho)}\right] \qquad (2.33)$$

where $S_3 = S_1 + S_2$. Letting $A_i = S_i/\sigma^2$, we can maximize $[E_{oi}(\rho,Q)-\rho R_i]$ over ρ to get the parametric equations

$$E_{ri}(R_i) = \frac{\rho_i^2 A_i}{2(1+\rho_i)(1+\rho_i+A_i)} \qquad 0 \le \rho_i \le 1$$

$$R_i = \frac{1}{2} \ln\left[1 + \frac{A_i}{1+\rho_i}\right] - \frac{\rho_i A_1}{2(1+\rho_i)(1+\rho_i+A_i)} \qquad (2.34)$$

For rates lower than those where $\rho_i = 1$,

$$E_{ri}(R_i) = \frac{1}{2} \ln\left[1 + \frac{A_i}{2}\right] - R_i$$

$$\text{for } R_i \le \frac{1}{2} \ln\left[1 + \frac{A_i}{2}\right] - \frac{A_i}{4(2+A_i)} \qquad (2.35)$$

As in (2.22) and (2.23), the random coding exponent $E_r(R_1,R_2)$ is the minimum of $E_{ri}(R_i)$ over $i = 1,2,3$. The region R divides into three subregions as shown in fig. 2.7 where $E_{ri}(R_i)$ for each i is dominant. As the rates decrease, the error probability of type 3 errors decreases more rapidly than that for type 1 and 2 errors, so that for small rates the bound is dominated by errors in source 1 or 2 but not both.

For a single input additive Gaussian noice channel, choosing a coding ensemble with the Gaussian distribution is not quite the best thing to do. The best distribution results from a shell constraint; that is, code words are chosen with a Gaussian distribution <u>conditional</u> on the resulting word having an energy very close to NS_i. This distribution (see section 7.4, [10]) yields the same exponent to P_e as the sphere packing bound for rates sufficiently close to capacity.

For a multiaccess channel, it seems reasonable to again consider a random coding ensemble using a shell constraint on each set of code words. From the genie interpretation of type 1 and 2 errors, we see that P_{e1} is upperbounded by the probability of error for the first set of code words with the additive Gaussian noise but without the second set of code words. Thus, for i=1,2, we have $P_{ei} \leq a_i N \exp[-NE_{ri}(R_i)]$, where from section 7.4 of [10], a_i is a constant and $E_{ri}(R_i)$ is given by:

$$E_{ri}(R_i) = \frac{A_i-\gamma_i}{2\beta_i} + \frac{1}{2}\ln(\beta_i-\gamma_i) \qquad (2.36)$$

$$\text{for } \frac{1}{2}\ln\left[(1/4)(2+A_i+\sqrt{4+A_i^2}\,)\right] \leq R_i \leq \frac{1}{2}\ln(1+A_i) \qquad (2.37)$$

where

$$\gamma_i = \frac{A_i(\beta_i-1)}{2}\left[\sqrt{1+\frac{4\beta_i}{A_i(\beta_i-1)}} - 1\right] \qquad (2.38)$$

$$\beta_i = \exp(2R_i) \qquad (2.39)$$

For R_i less than the lower limit in (2.37),

$$E_{ri}(R_i) = 1 - \beta_i + \frac{A_i}{2} + \frac{1}{2}\ln[\beta_i(\beta_i - \frac{A_i}{2})] - R_i \qquad (2.40)$$

$$\beta_i = \frac{1}{2}\left[1 + \frac{A_i}{2} + \sqrt{1 + \frac{A_i^2}{2}}\right] \qquad (2.41)$$

For rates satisfying (2.34), the sphere packing bound for the single input channel gives a lower bound

$$P_{ei} \geq \exp\left[-N[E_{ri}(R_i) + o(N)]\right] \qquad ;i=1,2$$

for all codes where $o(N)$ approaches 0 with increasing N.

For type 3 errors, the situation is less simple since the combined code words $\mathbf{x} + \mathbf{w}$ are not constrained. In fact, if, after constraining $\mathbf{x}$ to have energy NS_1 and $\mathbf{w}$ to have energy NS_2, we then constrained $\mathbf{x}+\mathbf{w}$ to have energy $N(S_1+S_2)$, we would then be constraining the code words of the two codes to be orthogonal, which is just a generalized version of the frequency division multiplexing discussed previously.

We now develop a bound on P_{e3} using a shell constraint on the code words $\mathbf{x}_m$ and $\mathbf{w}_\ell$. Choose each $\mathbf{x}$ independently using the density $Q_1(\mathbf{x})$ and each $\mathbf{w}$ using the density $Q_2(\mathbf{w})$ where

$$Q_i(\mathbf{x}) = \mu_i^{-1}\phi_i(\mathbf{x}) \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi S_i}} \exp\left[\frac{-x_n^2}{2S_i}\right] \qquad (2.42)$$

$$\phi_i(x) = \begin{cases} 1; & \text{for } NS_i - \delta < \sum_{n=1}^{N} x_n^2 \leq NS_i \\ 0; & \text{otherwise,} \end{cases} \tag{2.43}$$

where δ is an arbitrary positive number, and μ_i is a normalizing constant to make $Q_i(x)$ integrate to 1. Substituting (2.40) for $Q_1(x)$ and $Q_2(w)$ into (2.8), replacing sums with integrals, and upper bounding

$$\phi_i(x) \leq \exp[r_i\delta + \sum_{n=1}^{N} r_i(x_n^2 - S_i)] \tag{2.44}$$

For any $r_i \geq 0$, $i = 1,2$, we find that (2.8) breaks into a product form (as in section 7.3 of [10]). After some tedious integration, we get, for any ρ, $0 \leq \rho \leq 1$,

$$P_{e3} \leq \left[\frac{\exp[\delta(r_1+r_2)]}{\mu_1\mu_2}\right]^{1+\rho} \exp[-N(E_{03}(\rho,r)-\rho R_3)] \tag{2.45}$$

$$E_{03}(\rho,r) = (1+\rho)\ln\left[\frac{e\sqrt{\theta_1\theta_2}}{1+\rho}\right] - \left(\frac{\theta_1+\theta_2}{2}\right) + \frac{\rho}{2}\ln\left[1 + \frac{A_1}{\theta_1} + \frac{A_2}{\theta_2}\right] \tag{2.46}$$

$$\theta_i = (1+\rho)(1-2r_iS_i) \tag{2.47}$$

The first term in (2.45) is proportional to $N^{1+\rho}$ for any given choice of r_1, r_2, and δ, so we simply bound it by aN^2 for some suitable a. The exponent can be optimized over ρ, r_1, r_2 (or equivalently over ρ, θ_1, θ_2 for $0 \le \rho \le 1$, $0 \le \theta_i \le 1+\rho$). For the important case where $A_1 = A_2$, the optimization can be carried out explicitly. Here by symmetry, the optimal θ_1 and θ_2 are equal, and such a solution is also valid, but not optimal, for all A_1 and A_2. Using θ for θ_1 and θ_2 and A for A_1 + A2,

$$E_{03}(\rho,r) = (1+\rho)\ln\left(\frac{e\theta}{1+\rho}\right) - \theta + \frac{\rho}{2}\ln\left(1 + \frac{A}{\theta}\right) \qquad (2.48)$$

Optimizing the exponent, we find that for

$$\frac{1}{2}\ln\left[\frac{1}{2}\left(1 - \frac{A}{4} + \sqrt{1 - \frac{A}{2} + \frac{A^2}{4}}\right)\right] \le R_3 \le \frac{1}{2}\ln(1+A), \qquad (2.49)$$

$$E_{r3}(R_3) = (1 + \rho - \theta) + \ln\frac{\theta}{1+\rho} \qquad (2.50)$$

$$\theta = \frac{1+\rho-A}{2} + \frac{1}{2}\sqrt{(1+\rho)^2+A^2+2A} \qquad (2.51)$$

$$\rho = \left[\frac{1}{2} + \frac{2\beta}{A} - \frac{1}{2}\sqrt{1 + \frac{8\beta}{A} + \frac{16\beta}{A^2}}\right]^{-1/2} - 1 \qquad (2.52)$$

$$\beta = \exp(2R_3) \qquad (2.53)$$

For R_3 less than the lower bound in (2.49),

$$E_{r3}(R_3) = 2[\ln\frac{\theta}{2} - \theta - 1] + \frac{1}{2}\ln(1 + \frac{A}{\theta}) - R_3 \qquad (2.54)$$

$$\theta = 1 - \frac{A}{2} + \frac{1}{2}\ln(A^2+2A+4) \qquad (2.55)$$

This exponent lies roughly half way between the previously derived exponent without a shell constraint and the exponent with a shell constraint that would result for a single input Gaussian channel with signal to noise ratio A (i.e. that given by (2.36)-(2.41).

When we take the minimum of the three exponents $E_{ri}(R_i)$ for i = 1,2,3, we again find that the achievable region **R** breaks into 3 subregions, one where each bound is dominant; the regions look the same as in fig 2.7, although numerically they are somewhat different. We now know, however, that whenever the rate pair (R_1,R_2) is in $\mathbf{R}_1$ (or $\mathbf{R}_2$) and R_1 (or R_2) is above the critical rate of (2.36), then $E_r(R_1,R_2)$ is indeed the exponent for optimal

codes. For the symmetric case where $R_1 = R_2$, the region $\mathbf{R}_3$ vanishes for small enough $R_1 = R_2$, and if the point where $\mathbf{R}_3$ vanishes is above the critical rate for R_1 and R_2, then the optimum exponent is given by (2.37)-(2.39) between the point where $\mathbf{R}_3$ vanishes and the critical rate. This phenomen occurs whenever the combined signal to noise ratio A_3 is below about 3.

3. Coding Techniques

While the theoretical development of coding theorems for multiaccess channels is quite advanced, very little has been done with respect to general techniques for multiaccess coding. As pointed out in the introduction, what is needed is a coding technonology applicable where there are a large set of transmitters but only a small subset simultaneously use the channel while the others are idle. Here, however, we restrict ourselves to the simpler problem of the two input channel of fig. 2.1 where both sources always have something to send.

First we observe that the error probability bounds evaluated in the last section apply equally well to ensembles of linear codes. The argument for this is the same as in section 6.2 [10]. In general, binary linear codes can be generated for each transmitter, and sub-blocks of these binary digits can be mapped many to one into the channel input alphabet, thus achieving any desired relative frequency of utilization of the various input letters.

Random coding bounds for convolutional codes have also been generalized from single input channels to multiaccess channels [16] with the same type of large exponent as occurs for the single input channel. Thus there is no problem generating good codes, either block or convolutional. The problem, as with single input channels, is with decoding.

Before discussing decoding, a brief discussion of channel modelling is in order. The discrete time channels dear to the hearts of information theorists implicitly assume that

carrier phase and sampling time in physical channels are part of the channel model. Furthermore, ideal performance of these elements is usually assumed. For single input channels this separation is usually perfectly reasonable, but for multiaccess channels it is often questionable. For example, for the AWGN multiaccess channel, it is well known [17],[18] that feedback can increase the achievable R_1+R_2 beyond that achievable by a single source of rate R_1+R_2 and energy constraint S_1+S_2. In other words, the individual transmitters are limited to S_1 and S_2 respectively, but the signal energy at the receiver exceeds S_1+S_2. This means that the two transmitting antennas are acting as a phased array and that the additional receiver energy comes from antenna gain (along with very clever feedback coordination). While this is not impossible, it is certainly not a conventional situation.

Typically we should expect the received carrier phase from the one transmitter to be roughly independent of that from the other. Approximate baud synchronism between the transmitters is slightly more reasonable than phase synchronism and approximate block synchronism is eminently reasonable with only marginal feedback communication.

There appears to be little of a general nature that can be said about the effect of asynchronism between the sources at the phase and baud level. For the specific case of an AWGN channel, however, the situation is much simpler. Using a Gaussian ensemble (with or without a shell constraint) to generate code words, the discrete time code words of the last

section can be considered as time samples over the block period of a narrow band stationary Gaussian process with alternate letters representing in phase and out of phase components. Thus for a given set of randomly chosen waveform code words, a change of receiver carrier phase and sample time will change the discrete time code but will not change the ensemble statistics (aside from some end effects at the ends of the block which we ignore). The decoder must know the relative carrier phase and sample time for each of the two transmitters but there is no need for the two to be synchronized together. In summary, the discrete time AWGN multiaccess model of the last section is adequate for non-feedback communication maintaining only block synchronization, but is only adequate for feedback techniques in the rare case where the two transmitters are phase and baud synchronized.

The problem of lack of block synchronization for multiaccess channels is somewhat better understood than that of phase and baud synchronization. Assuming a discrete time model (i.e. assuming away the phase and baud synchronization problem), it has been shown [19] that with a bounded amount of uncertainty in timing between the transmitters, the feasible region **R** is the same as with perfect synchronization. Essentially one uses a coding constraint so large that the timing uncertainty becomes negligible. For complete uncertainty in timing, on the other hand, it has been shown [20] that the feasible region is the union region of fig. 2.4 rather than its convex hull. The essential idea here is that time sharing cannot be used in the

total absence of relative timing between the transmitters.

Having cautioned the reader about the modeling problems inherent in a discrete time memoryless model of multiaccess channels, we now return to this model to see what can be said about coding.

First, there is a fairly simple general approach that can reduce the decoding problem to several single source decoding problems. First suppose that (R_1,R_2) satisfies $R_1 < I(X;Y|W)$, $R_2 < I(W;Y)$ for some assignment $Q_1(x)$, $Q_2(w)$. Over the ensemble of codes using Q_1,Q_2, a decoder can decode the **w** code word by ignoring the **x** code word and assuming a single input channel with transition probabilities $P(y|w) = \Sigma_x\, Q_1(x)P(y|xw)$. Over the ensemble of codes for the first encoder, this is precisely the set of transition probabilities from w to y. Thus a "good" decoder for a single input channel can decode **w** reliably. Given **w**, another decoder for a single input channel can decode **x** using $P(y_n|x_nw_n)$. This second decoding is somewhat unconventinal for single inputs in that the transition probabilities depend on w_n and thus vary with n, but a number of decoding techniques such as sequential decoding and Viterbi decoding can deal with this situation.

As can be seen from fig. 3.1, any (R_1,R_2) in the interior of the achievable region of (2.3)-(2.5) for a given Q_1,Q_2 can be represented as a convex combination of two rate pairs, one of which, (R_1',R_2'), satisfies

$$R_1' < I(X;Y|W); \qquad R_2' < I(W;Y) \qquad (3.1)$$

and the other of which satisfies

$$R_1'' < I(X;Y); \qquad R_2'' < I(W;Y|X) \qquad (3.2)$$

Codes for each of these rate pairs can be decoded by the two step procedure described above and (R_1,R_2) can be decoded by time sharing between two such codes.

Finally, any point in the interior of the achievable rate region is a convex combination of two rate pairs, one of which satisfies (2.3)-(2.5) with strict inequality for some Q_1Q_2 and the other for some other $Q_1^*Q_2^*$. Thus an arbitrary point in the interior of **R** can be reliably decoded by time sharing between at most 4 codes, two of which use rates satisfying (3.1), (3.2) respectively for Q_1Q_2 and the other two of which satisfy (3.1), (3.2) for $Q_1^*Q_2^*$.

This approach is not entirely satisfactory for two reasons. The first is that the random coding exponents for error probability in this approach are often much smaller than those for joint decoding of the two code words together. If we use error exponents as a crude measure of decoding complexity, this indicates that the price of avoiding joint decoding is much greater complexity for the single input decoders. Note, however, that error exponents can sometimes be misleading as a guide to

decoding complexity. For example, the random coding exponent for a noiseless binary channel is not large, whereas coding and decoding are completely trivial.

The other objection to this approach is that it fails to provide much insight into the question of joint decoding of several sources. It certainly does not generalize to the use of a small but unknown subset of a large set of transmitters.

A second, simpler but less general, approach is to decode the code words from each transmitter independently regarding the other as noise. From fig. 2.5, it is seen that for the AWGN channel with small signal to noise ratio, the achievable rate region is almost rectangular. Analytically $I(X;Y) = (1/2)\ln[1 + A_1/(1+A_2)]$ which is close to $I(X;Y|W) = (1/2)\ln[1+A_1]$ when A2 is small. In this case, the error exponent for individual decoding is almost the same as for joint decoding.

This second approach can be carried one step further by choosing all the code words for transmitter 1 to be orthogonal or almost orthogonal to all those for transmitter 2. This is the approach taken in frequency division multiplexing, and has the added advantage of largely eliminating the problems caused by relative differences in carrier phase and baud timing between the transmitters.

This approach is also used in spread spectrum communication. This has the added advantage of allowing a large number of transmitters, all of whose code sets are approximately orthogonal to all the other code sets. When only a subset of the transmitters transmit at one time, the interference from the

other transmitters is reduced and the individual code words can be successfully decoded. This same approach has been used by Cohen et al [21] and Sommer [22] in the context of multiaccess pulse position modulation.

For an arbitrary discrete time memoryless multiaccess channel, perhaps with more than two transmitters, one can similarly investigate ways to choose code word sets for the individual transmitters in such a way that they are mutually non interfering (more precisely, so that they can be individually decoded with small error probability). Time sharing within a code word is one possibility, but depending on the channel, other possibilities might be preferable, as we have seen for the AWGN channel. A more difficult related problem is to choose the code word sets in such a way as to maintain the non-interference property in the presence of lack of baud synchronism between the transmitters. We have seen that this can be done for the AWGN channel, and Massey's coding scheme [14] for the asynchronous collision channel also achieves this objective; at present, however, no approaches are known for general discrete time memoryless channels.

As a third approach to decoding, consider true joint decoding of the two code words. I will not consider algebraic decoding techniques here since an algebraic structure must be matched in some sense to the channel characteristics and I am not aware of any interesting examples of general algebraic approaches for multiaccess channels. Viterbi decoding of convolutional codes is another possibility, but it does not appear very

promising as a joint decoding technique. The problem is that the decoder should track all possible states of both encoders, which leads to a combined number of states which is the product of the individual numbers of states. With more than two transmitters, the problem is even worse.

Finally, sequential decoding appears to be a general approach to multiaccess joint decoding and it has been shown [23] that lack of block synchronization is not a serious impediment to its operation. Unfortunately, at this time, it is not clear how to make sequential decoding work for a multiaccess channel. To explain the difficulty, recall that sequential decoding is a search procedure that hypothesizes the encoded sequence up to a given point and either proceeds forward by extending the encoded sequence or searches backward depending on the value of a "metric" that stochastically drifts upward when the decoder is following the actual uncoded sequence and drifts downward when the decoder gets off the track.

The problem, now, is that the decoder can go off the track in three ways, corresponding to the three types of errors in section 2. Unfortunately the appropriate metric to use depends on the type of error being made, and this knowledge is unknown to the decoder. It appears that no single metric is adequate for sequential decoding to work on a general discrete memoryless multiaccess channel with bounded expected computation up to the normal computational cut off rate. It might be possible to develop a sequential decoding algorithm that utilizes several metrics simultaneously, but so far no such algorithm has

been devised.

Another fundamental problem with sequential decoding has recently been discovered by Arikan [24]. Arikan considers a multiaccess binary erasure channel where X = {0,1}, W = {0,1} and Y = {(0,0), (0,1), (1,0), (1,1), (e,e)}. With probability 1-ε, for some ε > 0, y = (x,w), whereas with probability ε, independent of the input, y = (e,e). In effect we have two erasure channels with perfectly correlated erasures. Using equiprobable inputs for each transmitter, we can formally calculate the computational cutoff region $\mathbf{R}_{comp}$ for a joint decoder as

$$R_1 \leq E_{o1}(1,Q) = -\ln\left[\frac{1+\varepsilon}{2}\right] \qquad (3.3)$$

$$R_2 \leq E_{o2}(1,Q) = -\ln\left[\frac{1+\varepsilon}{2}\right] \qquad (3.4)$$

$$R_3 \leq E_{o3}(1,Q) = = \ln\left[\frac{1+3\varepsilon}{4}\right] \qquad (3.5)$$

we note that

$$-2\ln\left[\frac{1+\varepsilon}{2}\right] > -\ln\left[\frac{1+3\varepsilon}{4}\right]; \quad \text{all } \varepsilon,\ 0 < \varepsilon < 1 \qquad (3.6)$$

Thus for $R_1 = R_2$, (3.5) is the active constraint, and even without any of the metric problems discussed above, (3.5) limits the achievable rate with joint sequential decoding. However using separate sequential decoders for the two transmitters and ignoring the erasure correlation, we can achieve the higher rates of (3.3) and (3.4).

To make the situation worse, we see that $-\ln[(1+3\varepsilon)/4]$ is also the computational cut off rate of a single input quaternary erasure channel. However, by regarding the inputs to the quaternary channel as two binary digits and using separate convolutional encoders and decoders for the two digits, we can again achieve the higher rates. The difficulty here does not reside in the particular search algorithm being used. Over the ensemble of convolutional codes for the quaternary input (or pairs of codes for binary inputs), the expected number of potential encoded sequences (or pairs of sequences) at length N which are as likely as the transmitted sequence (or pair) is exponentially increasing in N for any combined rate in excess of $-\ln[(1+3\varepsilon)/4]$. The conclusion that one must reach is that R_{comp} is not really a fundamental parameter of communication. This same example, in the context of the photon channel, has been discussed by Massey [25] and Humblet [26].

Summarizing the previous approaches to decoding, we see that much more research is necessary before any cohesive body of knowledge about coding and decoding for multiaccess channels will exist.

4) COLLISION RESOLUTION

As briefly discussed in the introduction, the collision resolution approach to multiaccess communication focuses on allocating the multiaccess channel among a large set of users at different transmitting sites. It has the weakness, however, of essentially ignoring the communication aspects of the problems. We start by a set of assumptions that limit the class of systems we will be considering.

a) Slotted System: We assume that each message (packet) to be transmitted requires one time unit (a slot) for transmission. All transmitters are synchronized so that all transmissions start at an integer time and end before the next integer time. Such synchronization is usually not too difficult given stable clocks and given a small amount of timing feedback from the receiver. In case of propagation delays, the timing is relative to the receiver, so that each packet starts to arrive at the receiver at an integer time. Naturally some guard space is required in practice, but we neglect that here. Note that this assumption precludes both the possibility of sending short packets to make reservations for long packets and of carrier sensing, which we discuss later. Such systems can be understood much more simply after this basic model is understood.

b) Collision or Perfect Receiption: We assume that if more than one transmitter sends a packet in a slot (the time from one integer to the next), then there is a collision and the receiver gets no information about the contents or origins of the transmitted packets. If just one transmitter sends a packet in a

slot, it is received with no errors. This is the assumption that removes the noise and communication aspects from the problem; it allows collision resolution to be studied in the simplest context but also severely limits the class of strategies and tradeoffs that can be considered.

c) Infinite Set of Transmitters: Assume that each arriving packet arrives at a transmitter that has never previously received a packet. This precludes queueing at individual transmitters and precludes the use of TDM. This is an unreasonable assumption from a practical point of view, but note that given any algorithm determining when the transmitters send packets, a finite set of transmitters can use the same algorithm regarding each packet arrival as corresponding to a separate conceptual transmitter. In this case, a real transmitter will sometimes send multiple packets at the same time, causing a collision. This shows, first that assumption c) provides a worst case bound on a finite set of transmitters and second, that the difference is only significant when two or more packets are waiting at the same transmitter. Collision resolution algorithms are primarily useful to reduce delay over what would be achieved with TDM, so in this low delay region, having multiple packets at a transmitter should be relatively rare and the performance with a finite set of transmitters shuld be well approximated by the performance with an infinite set. The major advantage of the infinite set assumption is that we can use the maximum throughput of an algorithm as a qualitative measure of the goodness of the algorithm without allowing for the somewhat incidental

improvements of throughput that could be achieved when transmitters have multiple packets to send.

d) Poisson Arrivals: Assume that new packet arrivals are Poisson at an overall rate λ. Given assumption 3, no other arrival process would make much sense.

e) 0, 1, c Immediate Feedback: Assume that by the end of each slot, each transmitter learns whether 0 packets, 1 packet, or more than one packet (c for collision) were transmitted in that slot. This is the only information that each transmitter gets about the existence of packets elsewhere. The assumption of immediate feedback is often unrealistic, but collision resolution algorithms can usually be easily modified to deal with delayed feedback; the introduction of delay in the feedback, however, seems to greatly complicate analysis with no apparent benefit in insight. The assumption of 0, 1, c feedback implies that the receiver (or the transmitters themselves) can distinguish between an idle channel and a collision, which is not always reasonable. It also implies that idle transmitters are always listening for this feedback, which is not always desirable. Some alternative forms of feedback will be discussed in what follows.

4.1 SLOTTED ALOHA: The simplest form of collision resolution strategy using the assumptions above is Slotted Aloha, due to Roberts [27]. Slotted Aloha is a variation of pure Aloha, devised by Abramson [1], which will be briefly discussed subsequently. In slotted Aloha, whenever a packet arrives at one of the transmitters, that packet is transmitted in the next slot. Whenever a collision occurs in a slot, each packet involved in

the collision is said to be backlogged and remains backlogged until it is successfully transmitted. Each such backlogged packet is transmitted in each slot with some fixed probability $p > 1$, independent of past slots and of other packets. Note that if p were 1, backlogged packets would continue colliding and no more packets would ever be successfully transmitted. Note also that because of the effectively infinite set of transmitters, the collision cannot be resolved by transmitters waiting some number of slots determined by the identity of the transmitter. Such strategies can be used with a known set of transmitters and can be made to behave like TDMA under heavy loading.

It can easily be seen that slotted Aloha can be analyzed as a homogeneous Markov chain, using the number of backlogged packets at each integer time t as the state. The state at time t includes packets that collided in the slot from t-1 to t but does not include new packet arrivals from t-1 to t. Let k be the state at time t and k+i be the state at t+1. Note that i can never be less than -1 (i.e. at most one backlogged packet can be successfully transmitted in the slot [t,t+1]. Furthermore, $i = -1$ if no new packets arrived in [t-1,t) and exactly one backlogged packet is transmitted in [t,t+1). This event has probability $kp(1-p)^{k-1}e^{-\lambda}$. The state stays the same (i=0) either if no new packets arrived in [t-1,t) and no backlogged packet is successfully transmitted in (t,t+1) or if one new packet arrived in [t-1,t) and is successfully transmitted in [t,t+1)

Analyzing the cases $i > 0$ in the same way, we see that the state transition probabilities $P_{k,k+i}$ are given by

$$P_{k,k+i} = \begin{cases} kp(1-p)^{k-1}e^{-\lambda} & i = -1 \\ [1-kp(1-p)^{k-1}]e^{-\lambda} + (1-p)^{k}\lambda e^{-\lambda} & i = 0 \\ [1-(1-p)^{k}]\lambda e^{-\lambda} & i = 1 \\ \frac{\lambda^{i} e^{-\lambda}}{i!} & i \geq 2 \end{cases} \quad (4.1)$$

In understanding how this chain behaves, we look first at the drift, D_k, defined as the expected value of i conditional on k (i.e. the expected difference between the state at t+1 and that at t conditional on the state at t).

$$D_k = \lambda - [(1-p)^{k}\lambda e^{-\lambda} + kp(1-p)^{k-1}e^{-\lambda}] \quad (4.2)$$

The first term λ is the arrival rate and the second term is the departure rate or throughput. Note that for any $\lambda > 0$ and any $p > 0$, D_k will be positive for all sufficiently large k. This means that if the system becomes sufficiently backlogged, it drifts in the direction of becoming more and more backlogged; this should not be surprising since collisions occur on almost all slots when the backlog gets sufficiently large. Kaplan [28] gives a simple but elegant proof that this type of chain is unstable (i.e. non-ergodic).

Despite the instability of slotted Aloha, it can still be a useful collision resolution approach especially if the system is modified to avoid or recover from the heavily backlogged state. Using a small value of p helps postpone the onset of the

catastrophic behavior above, and for small p, (4.2) can be well approximated by

$$D_k \approx \lambda - (\lambda+pk)e^{-(\lambda+pk)} \qquad (4.3)$$

Fig. 4.1 illustrates this equation. For $\lambda > e^{-1}$, we see that $D_k > 0$ for all k. For $\lambda < e^{-1}$, there is a range of k for which $D_k < 0$, and the size of this range increases as λ decreases and as p decreases. Unfortunately, λ is the arrival rate which we would rather not decrease, and small p means large delay between retrials of a collided packet.

This tradeoff in p is very undesirable; large p makes it very easy to enter the unstable heavily backlogged region, whereas small p causes large delay for collided packets in the stable region. The engineering solution is almost obvious--change p as the backlog k changes. Ideally, we would like to adjust p to keep $\lambda + pk = 1$, thus maintaining a throughput of e^{-1} for all $k > 0$. This keeps delay small when the backlog is small and keeps the system stable if $\lambda < e^{-1}$. The problem with this solution is that k is unknown, and either k must be estimated from the feedback or an apropriate value of p must be estimated. Hajek and VanLoon [29] have analyzed a class of algorithms in which p is updated at each slot simply as a function of the previous p and the feedback information. They showed that such functions cam be chosen for any $\lambda < e^{-1}$ so as to make the resulting system stable.

From (4.3), we see that D_k is positive whenever $\lambda > 1/e$. This is only an approximation of (4.2), but the approximation is

good when p is small, and p must be small when k is large to minimize D_k. Thus, for $\lambda > 1/e$, D_k is positive for all sufficiently large k no matter how p is chosen, so that slotted Aloha is unstable in this case even if k is known.

In the next subsection we show that much higher throughputs, and presumably smaller delays, are possible when newly arriving packets are sometimes held up and collisions are resolved in more sophisticated ways. The primary advantage that slotted Aloha has over these more sophisticated strategies is that slotted Aloha does not require all the feedback information we have assumed. For many physical multiaccess channels, particularly dispersive fading channels, it is difficult to distinguish an idle slot from a collision with high reliability. It is usually straightforward, through use of a cyclic redundancy check, to distinguish a successful transmission from idle or collision, and it can be seen that this kind of feedback is sufficient for slotted Aloha but not sufficient for the more sophisticated strategies. Unfortunately it is much more difficult to estimate the backlog with this type of feedback and it is an open research problem to determine whether slotted Aloha can be stabilized in this case.

Pure Aloha [1] was the precursor of slotted Aloha and avoids our assumption of a slotted system, although we continue to assume that each packet requires one time unit for transmission, that overlapping packets collide, and that assumptions c), d), and e) hold. Each newly arrived packet is transmitted immediately upon arrival and backlogged packets are transmitted

after a geometrically distributed delay. The probability of collision is higher here than in a slotted system; a packet starting transmission at time t will collide with other packets starting anywhere in the interval (t-1,t+1). The upper bound on throughput becomes $(2e)^{-1}$ and the same kinds of stability issues arise as for the slotted system. A major practical advantage of pure Aloha, however, is its ability to handle packets of different lengths [30,31].

4.2 SPLITTING ALGORITHMS: In our discussion of slotted Aloha, we saw that the throughput is upper bounded by 1/e regardless of the strategy used to adjust the retransmission probability of collided packets. This bound was imposed by the restriction that new arrivals were always transmitted in the next slot after their arrival and that backlogged packets depended upon a single parameter p for retransmission. To get an intuitive idea of why the transmission of new arrivals should sometimes be postponed, consider a slot in which two packets collide. If the new arrivals were held up until the collision were resolved, then a reasonable strategy would be for each colliding packet to retransmit in the following slot with probability 1/2. With probability 1/2, then, a successful transmission occurs and the other packet would be transmitted in the following slot. Alternatively, with probability 1/2, another collision or an idle slot ensues, wasting one slot. Again, in this case, each packet would be transmitted in the following slot independently with probability 1/2, and so forth until the two packets are successfully transmitted. The expected number of slots required

to successfully transmit the two packets is easily seen to be 3, which yields an effective throughput of 2/3 during the collision resolution period.

This concept of probabalistically splitting the set of packets involved in a collision into a transmitting set and a non-transmitting set while making other packets wait is the central idea of a variety of collision resolution algorithms that achieve throughputs larger than 1/e while using assumptions a) to e); we call these algorithms splitting algorithms. These algorithms differ in the rules used for splitting the collision set (which might involve more than two packets) and in the rules for allowing waiting packets not involved in a collision to transmit after the collision is resolved.

The first splitting algorithms were the tree algorithms developed by Capetanakis [32], Hayes [33], and Tsybakov and Mikhailov [34]. In these algorithms, the system alternates between two modes--normal mode and collision resolution mode. When a collision occurs in normal mode, all transmitters go into collision resolution mode, all new arrivals wait until the next transition into normal mode, and all packets involved in the collision independently select one of two subsets with equal probability. We view each subset as corresponding to a branch from the root of a rooted binary tree (see fig. 4.2). In the slot following the collision, the first of these subsets is transmitted. If another collision occurs, this subset is further split into two smaller subsets, corresponding to further branches growing from the original branch. The first of these

subsets is transmitted in the the next slot, and if this transmission is successful or idle, the second of the subsets is transmitted in the following slot. In general, whenever the transmission of a subset results in a collision, the subset is split and two new branches of the tree are grown from the old branch. Whenever the transmission of a subset is idle or successful (i.e. the subsetis empty or contains one packet), the next slot is used to transmit the next subset. When all subsets have been exhausted, the normal mode is again entered.

It should be apparent that if this algorithm spends many slots resolving a collision, then typically many new arrivals will eagerly be awaiting the return to normal mode and a resounding collision will ensue. What is even worse is that many successive collisions will follow until the expected number of pakets in a subset becomes on the order of 1. Thus the algorithm can be improved by eliminating the normal mode; at the end of a collision resolution period, a new collision resolution period is immediately entered and each waiting packet randomly joins one of k subsets. The number k is chosen as a function of the length of the preceding collision resolution period so that the expected number of packets per subset is slightly more than one. Thus the corresponding tree has k branches rising from the root and two branches rising from each non-leaf node.

Capetanakis [32] showed that this algorithm has a maximum throughput of 0.43 and is stable for all input rates less than 0.43. The maximum throughput attainable with tree algorithms was later increased to 0.46 due to a simple improvement first suggested by Massey [35]. Note what the algorithm does when the set involved in a collision is split into two subsets of which the first is empty. The first slot following the collision is then idle and the next is a collision, involving all the packets in the first collision. Massey's improvement was to avoid this predictable collison by resplitting the second subset of a collision set whenever the first subset is empty.

The next improvement in throughput was due to Gallager [36], and somewhat later with a more complete analysis, by Tsybakov and Mikhailov [37]; this involved eliminating the tree structure entirely. We shall describe this algorithm precisely later, since it is considerably easier to analyze than the tree algorithm. First, however, we view it as another modification of the tree algorithm. With a little thought, one can see that the number of packets in a subset that has had a collision is a Poisson random variable conditional on the number being 2 or more. If the packets in this set are randomly divided into two subsets, then it can also be seen that if the first subset contains 2 or more packets (i.e. another collision) then, conditional on this, the number of packets in the second subset is Poisson. Thus, as far as the algorithm is concerned, this subset is statistically identical to some time interval of new

arrivals, and the algorithm would be improved if, rather than wasting a slot on this subset, we simply treated it like waiting new arrivals. We will get to the bookkeeping issue of how to do this shortly, but note that if we eliminate the second subset as a separate entity every time the first subset is divided, then we never have more subsets to consider than we started with.

The easiest way to do the bookkeeping concerning subsets and waiting packets is by means of the arrival times of the packets. If all the packets that arrived in a given time interval are transmitted in a slot and a collision results, then the interval is split into two equal subintervals and the packets in the first subinterval are regarded as the first subset and those in the second as the second subset. With this approach, packets are always sent in a first come first served (FCFS) order, so we call this a FCFS splitting algorithm.

We now express the algorithm precisely. Suppose that at integer time t the algorithm has successfully transmitted all packets that arrived before some time T (not necessarily integer). In the slot [t,t+1), all the packets that arrived between T and T+μ are transmitted. The parameter μ is determined by all each transmitter based on the history of the feedback up to time t. The transmitters also calculate T based on the feedback history. It is helpful to view the packet arrivals in [T,t) as being in a distributed queue (see fig. 4.3). We would like to allocate the queued packets one at a time starting at the front of the queue, but the individual arrival times are unknown except that each transmitter containing a packet knows that

packet's arrival time. Thus the algorithm attempts to allocate an interval μ at the front of the queue for the next slot so as to transmit the waiting packets as quickly as possible. Note that maximizing the probability of success in the next slot is not the best thing to do since, as we have seen, a collision in the next slot allows a higher throughput in the succeeding few slots than is possible with an idle slot or successful slot.

The algorithm given below determines the allocation interval $\mu(t)$ and head of queue time $T(t)$ for the slot $[t,t+1)$ in terms of the allocation interval $\mu(t-1)$, head of queue $T(t-1)$, and the feedback $(0,1,c)$ for the slot $[t-1,t)$. There is also a binary state $Q(t)\varepsilon(1,2)$ which is a function of $Q(t-1)$ and the feedback for $[t-1,t)$. The state $Q(t-1)$ also enters into the determination of $\mu(t)$ and $T(t)$. $Q(t)$ is set to 2 if the interval used in slot $[t-1,t)$ has been divided by 2 for slot $[t,t+1)$ and is 1 otherwise. Thus $Q(t)$ is the number of subsets currently under consideration. The algorithm also has a parameter μ_0 that determines the size of allocation interval to be used after a collision resolution period is completed. For maximum throughput, μ_0 turns out to be 2.6. Note that the allocation interval is also limited by $t-T(t)$, the interval of arrival times that are still waiting for transmission.

FCFS Splitting Algorithm:

if feedback = c then
$T(t) = T(t-1)$; $Q(t) = 2$;
$\mu(t) = \mu(t-1)/2$ (4.4)

if feedback = 0 or 1 and $Q(t-1) = 1$ then
$T(t) = T(t-1)+\mu(t-1)$; $Q(t) = 1$;
$\mu(t) = \min[\mu_0, t-T(t)]$ (4.5)

if feedback = 1 and $Q(t-1) = 2$ then
$T(t) = T(t-1)+\mu(t-1)$; $Q(t) = 1$;
$\mu(t) = \mu(t-1)$ (4.6)

if feedback = 0 and $Q(t-1) = 2$ then
$T(t) = T(t-1)+\mu(t-1)$; $Q(t) = 2$;
$\mu(t) = \mu(t-1)/2$ (4.7)

In case of a collision in slot $[t-1,t)$, Eq. (4.4) splits the allocation interval $[T(t-1), T(t-1)+\mu(t-1))$ into two intervals $[T(t-1), T(t-1)+\mu(t-1)/2)$ and $[T(t-1)+\mu(t-1)/2, T(t-1)+\mu(t-1))$. $Q(t) = 2$ allows the algorithm to "remember" the existence of these two subintervals. If there was a previous subinterval from $[T(t-1)+\mu(t-1), T(t-1)+2\mu(t-1))$, the algorithm "forgets" about it at this point, regarding that subinterval as part of the waiting queue. Given two or more packets in $[T(t-1), T(t-1)+2\mu(t-1))$ and two or more packets in $[T(t-1), T(t-1)+\mu(t-1))$, the number of packets in $[T(t-1)+\mu(t-1), T(t-1)+2\mu(t-1))$ is easily seen to be Poisson with parameter $\lambda\mu(t-1)$.

Eq. (4.5) corresponds to the end of a collision resolution period or a subsequent period with no collisions and simply moves the head of the queue and allocates a new interval. Eq. (4.6) corresponds to a successful transmission of the first subinterval from a previous collision and movement to the second subinterval. Finally (4.7) corresponds to Massey's improvement on the tree

algorithm when a collision followed by an idle (or perhaps seveal idles) is followed by splitting the second subinterval.

The FCFS splitting algorithm can be analyzed as a homogeneous Markov chain, using $Q(t)$, $\mu(t)$ and $t-N(t)$ as the state for integer values of t. It is simpler, however, to analyze a single collision resolution period, starting at a t for which $\mu(t) = \mu_0$ and $Q(t) = 1$ and ending immediately before the next t for which $\mu(t) = \min(\mu_0, N(t)-t)$ and $Q(t) = 1$. The resulting Markov chain is then independent of $N(t)-t$ (aside from the initial assumption that $N(t)-t \geq \mu_0$) and allows us in principle to find the distribution of the number of slots and number of successful transmissions in a collision resolution period. Note that in each update of μ (aside from the beginning of the collision resolution period), μ either stays the same or is divided by 2, so that $\mu = 2^{-i}\mu_0$ in all cases for some integer $i \geq 0$. Thus each state of the chain can be represented as $S_{j,i}$ where $j = Q(t) \varepsilon (1,2)$ and i is such that $\mu(t) = 2^{-i}\mu_0$. In state $S_{2,i}$ $(i>1)$, the only possible transitions are to $S_{2,i+1}$ if an idle or collision occurs or to $S_{1,i}$ if a success occurs. From $S_{1,i}$, $i \geq 0$, the only possible transitions are to $S_{2,i+i}$ if a collision occurs or to $S_{1,0}$ (representing the end of the period) otherwise (see fig. 4.4).

All that remains to complete the chain is to calculate the transition probabilities, $P_{2,i}$ for a transition from $S_{2,i}$ to $S_{1,i}$ and $P_{1,i}$ for a transition from $S_{1,i}$ to $S_{1,0}$. In state $S_{2,i}$, we have two subintervals each of size $\mu_i = \mu_0 2^{-i}$. The number of packets in each subinterval is a Poisson random variable with

parameter $\lambda\mu_i$ conditional on the sum of the number of packets in the two subintervals being two or more. The transition to $S_{1,i}$ occurs if the first subinterval contains exactly one packet (i.e. the transmission of the first subinterval is successful). The probability of this is then

$$P_{2,i} = \frac{\lambda\mu_i e^{-\lambda\mu_i}[1-e^{-\lambda\mu_i}]}{1 - e^{-2\lambda\mu_i}(1+2\lambda\mu_i)} \quad ; \; i \geq 1 \qquad (4.8)$$

In state $S_{1,i}$, $i \geq 1$, we are about to transmit the second of two subintervals each of size μ_i. The number of packets in each subinterval is Poisson with parameter $\lambda\mu_i$ conditional both on the sum being two or more and the first interval containing exactly one packet. This means that the number of packets in the second subinterval is Poisson conditional on one or more packets in the second subinterval. The probability of a transition to $S_{1,0}$ is then the probability of exactly one packet, so

$$P_{1,i} = \frac{\lambda\mu_i e^{-2\mu_i}}{1 - e^{-\lambda\mu_i}} \quad ; \; i \geq 1 \qquad (4.9)$$

Finally the probability of a direct transition from $S_{1,0}$ to $S_{1,0}$ is

$$P_{1,0} = (1 + \lambda\mu_0)e^{-\lambda\mu_0} \qquad (4.10)$$

It is now straightforward, by computer iteration, to find

the probability that each state is entered, starting at $S_{1,0}$, before the first return to $S_{1,0}$. Noting that the successful transmissions correspond to the transitions from $S_{2,i}$ to $S_{1,i}$ for each $i \geq 1$, transitions from $S_{1,i}$ to $S_{1,0}$ for $i \geq 1$, and successful transmissions directly from $S_{1,0}$ to $S_{1,0}$, we can calculate the expected number of successful transmissions and the expected number of slots per collision resolution interval. It turns out that the ratio of these two expected values is less than λ for all $\lambda < 0.4871$.

It can be seen from (4.9) and (4.10) that the probability of reaching $S_{2,i}$, given that $S_{2,i-1}$ has been reached, tends to 1/2 as i increases. Thus the number of slots and the number of successful transmissions in a collision resolution interval both have moment generating functions. From this, we can see that for any starting value of t-T(t) and any $\lambda < 0.4871$, the number of slots required to reach the end of a collision resolution interval where $t-T(t) < \mu_0$ also has a moment generating function and thus the algorithm is stable for $\lambda < 0.4871$.

The expected delay for this algorithm is considerably harder to analyze than the maximum throughput. Tsybakov and Likhanov [38] have found an upper bound on delay and more recently Huang and Berger [39] have constructed tight upper and lower bounds as well as simulation results. The expected delay is about 5 1/2 slots at $\lambda = 1/e$ and about 16 slots at $\lambda = 0.46$.

The FCFS splitting algorithm can be improved somewhat if the intervals are split in an optimal way after collisions. Because of the possibility of more than two packets in a collision, equal

subintervals are not quite optimal. Mosely and Humblet [40] and Tsybakov and Mikhaslov [37] show that choosing the optimum subintervals increases the maximum throughput to 0.4878. Recently another improvement of 3.6×10^{-7} has been made by Vvedenskaya and Pinsker [41]. Although this gain is small, it is of theoretical interest since it departs from the principal of always resolving one collision before trying any new intervals.

Considerable effort has been spent on finding upper bounds to the maximum throughput that can be achieved using the assumptions a) to e) [42, 43, 44, 45, 46]. The tightest bound known is 0.587 an is due to Mikhailov and Tsybakov [46]. Pippenger's result [42] is also of particular interest since he shows that if the amount of feedback is increased to give the number of packets involved in each collision, then any throughput up to one may be achieved.

One negative aspect of FCFS splitting algorithms (and also Massey's improvement on the Tree algorithms) is their susceptibility to noisy feedback. If an idle slot is mistakenly fed back to the transmitters as a collision, then the algorithm as stated will forever continue to split a smaller and smaller second subinterval. This problem could be solved, of course, by only splitting a given number of times in a row on receipt of 0 feedback and then trying the entire interval. The general subject of noisy feedback is still not well understood, but a number of partial results are known [35, 47, 48]. The review paper by Tsybakov [48] also reviews many variations on collision resolution algorithms for a variety of assumptions.

One interesting approach to noisy feedback and other variations from the ideal model above is that of developing algorithms that are as simple and robust as possible. Mathys and Flajolet [49] have recently developed an algorithm with an attractive tradeoff between simplicity and throughput. Newly arriving packets are always transmitted in the slot after their arrival, and backlogged packets use a ternary tree algorithm. Massey's improvement on the tree algorithm is not used, thus avoiding the above deadlock problems with noisy feedback;. The resulting maximum stable throughput is 0.40. It is rather surprising that a throughput greater than 1/e is possible while always allowing new arrivals to transmit in the next slot.

For multiaccess systems with a finite number of users, it is also of interest to modify these splitting algorithms so as to take advantage of the finite number of transmitters and to make a graceful transition from collision resolution to TDMA as the arrival rate increases. Specific approaches to this are discussed in [50,51]. The approach in [51] is also of interest because of drawing a parallel between splitting algorithms and group testing, as developed in the statistics community in the 40's and 50's.

4.3 CARRIER SENSING: We now want to change the basic assumptions a) to e). Note that in many multiaccess systems such as local networks, the transmitters can hear whether the other transmitters are sending anything. In such a situtation, it makes sense to give up the strict slotting specified in assumption a), and assume instead that a transmitter can start to send a packet in the middle of a data slot if no other transmitters are currently sending. This change is far more important than simply allowing idle slots to be used more efficiently, since now packets can start at different minislot times, thus avoiding many collisions.

Let α be the time required for all sources to determine that nothing is being transmitted; ie. α is the sum of the maximum propagation delay between sources and the time required by a receiver to reliably distinguish between signal and no signal. Assume that α time units after the beginning of a slot, if nothing is being transmitted in that slot, then the slot terminates and a new slot begins. Thus idle slots (sometimes called minislots) last for α time units and slots with one or more packets last for 1 time unit as before. We still assume that all packets require one time unit for transmission, that feedback is instantaneous at the end of a slot, that arrivals are Poisson with intensity λ, and that there are effectively an infinite number of sources. We first modify slotted Aloha for this new situation and then modify the FCFS splitting

algorithm. These techniques are called carrier sense multiple access (CSMA), although they do not imply the use of a carrier but simply the ability to quickly recognize the use of the channel by another transmitter.

We can model this situation in almost the same way as before. The only difference is that idle slots now last for a duration α, whereas successful and collision slots each last for one unit of time. For slotted Aloha, if a new packet arrives at a transmitter when an idle minislot is in progress, the packet begins transmission at the end of that minislot (thus turning the next slot into a full slot). If a transmission is in progress, the packet is regarded as a backlogged packet and begins transmission with some given probability p after each idle minislot. This technique was called non-persistent CSMA in the original description [52]; in an inferior, persistent, variaton, all transmission attempts during a busy slot would simply be transmitted at the end of that slot, thus causing a collision with a rather high probability. We ignore this alternative form in what follows.

To analyze CSMA, we can use a Markov Chain again, using the number of backlogged packets as the state and the ends of minislots as the state transition times. Rather than write out the state transition equations, which are not particularly insightful, we simply modify the drift in (4.2) for this new model. The expected number of arrivals in the minislot before the transition is $\lambda\alpha$, and with probability

$e^{-\lambda\alpha}(1-p)^k$, this minislot is followed by a full slot with λ expected arrivals. Note that there is always an unused minislot at the end of each full slot, but the arrivals in that minislot are considered as part of the following transition. The model could be changed to eliminate this wasted minislot, but the difference is negligible for small α. The expected number of departures per state transition is simply the probability of a success. Thus

$$D_k = \lambda\alpha + \lambda[1-e^{-\lambda\alpha}(1-p)^k] - [\lambda\alpha+pk/(1-p)]e^{-\lambda\alpha}(1-p)^k \tag{4.11}$$

This is minimized over p at

$$p = \frac{1 - \lambda(1+\alpha)}{k - \lambda(1+\alpha)} \tag{4.12}$$

The stability issues with CSMA slotted Aloha are almost the same as with ordinary slotted Aloha. One can control p by monitoring the feedback, or one can simply operate at a small value of λ and p and hope that the backlog never becomes too large. If we use the optimal value of p for each k, and substitute this in (4.11), we find that D_k is negative for all k so long as

$$\lambda(1+\alpha) \leq e^{-1+\lambda} \tag{4.13}$$

By expanding this in a power series for small α, we find that the system is scable for all λ less than $\sqrt{2\alpha}$. The optimal value of p then satisfies $pk \approx \sqrt{2\alpha}$. It is

interesting to observe that this optimal point occurs where the time spent on idle minislots is approximately equal to that spent on collisions; naturally there are many more idle slots than collisions, but idle slots have a much shorter duration. Delays also tend to be much smaller in a CSMA system since backlogged packets get a transmission opportunity every minislot, and, although the probability of transmitting in a minislot decreases with $\sqrt{\alpha}$, the probability of transmitting per unit time increases as $1/\sqrt{\alpha}$.

Next consider CSMA with pure Aloha. We will not analyze this in detail, but note that with the same carrier sensing time α and the same transmission probability p, the probability of collision increases by a factor of 2. This means that p should be decreased by a factor of $\sqrt{2}$ for maximum throughput, and thus the unslotted system has a maximum throughput of $1-2\sqrt{\alpha}$ for small α. We see that the difference between pure and slotted Aloha for CSMA is quite small for small α, and the synchronization required for slotting with CSMA is somewhat trickier than that for ordinary slotted Aloha. Thus pure Aloha appears to be the natural choice with CSMA.

Finally consider the FCFS splitting algorithm modified for CSMA. The same algorithm as in (4.4) to (4.7) can be used, although the parameter μ_0 should be changed, and as we shall see shortly, intervals with collisions should not be split into equal subintervals. Since collisions waste much more time than idle minislots, the basic allocation interval

μ_0 should be chosen very small. This means in turn that collisions with more than two packets are negligible in the analysis of the algorithm, and thus the analysis is much simpler than before.

As before we find the expected time and the expected number of successes in a collision resolution period, including a single idle or successful slot as a degenerate case of a collision resolution period. Let $\varphi = \lambda\mu_0$. With probability $e^{-\varphi}$, an original allocation interval is empty, yielding a collision resolution time of α with no successes. With probability $\varphi e^{-\varphi}$, there is an initial success, yielding collision resolution time $1+\alpha$ (as before, we include an empty minislot at the end of each full slot). Finally, with probability $(\varphi^2/2)e^{-\varphi}$, there is a collision yielding a collision resolution time of $1+T$, for some T to be calculated later, and two successes. Thus,

$$E(\text{time/period}) \simeq \alpha e^{-\varphi} + \varphi(1+\alpha)e^{-\varphi} + (1+T)(\varphi^2/2)e^{-\varphi} \qquad (4.14)$$

$$E(\text{packets/period}) \simeq \varphi e^{-\varphi} + 2(\varphi^2/2)e^{-\varphi} \qquad (4.15)$$

Note that we have used the approximation that only two packets occur in collisions here. As before, the maximum throughput that can be achieved is the ratio of (4.15) to (4.14),

$$\lambda_{max} \simeq (\varphi + \varphi^2)/[\alpha + \varphi(1+\alpha) + (\varphi^2/2)(1+T)] \qquad (4.16)$$

We can now maximize the right hand side of (4.16) over φ (ie. over μ_0). In the limit of small α, we get the asymptotic expressions

$$\varphi \simeq \sqrt{2\alpha/(T-1)} \tag{4.17}$$

$$\lambda_{max} \simeq 1 - \sqrt{2\alpha(T-1)} \tag{4.18}$$

Finally we must calculate T, the time to resolve a collision after it has occurred. Let x be the fraction of an interval used in the first subset when an interval is split. T includes the time α for the idle minislot that always follows a collision. If the next minislot is idle, α is the duration of the minislot, and $T-\alpha$ is the expected time still remaining to resolve the collision. Similarly, if another collision occurs, $1+T$ is the expected time for resolution. Finally, if a successful transmission occurs, $2(1+\alpha)$ is the required time for resolution. Thus

$$T \simeq \alpha + (1-x)^2T + x^2(1+T) + 4x(1-x)(1+\alpha) \tag{4.19}$$

T is minimized by $x = \sqrt{\alpha+\alpha^2} - \alpha$, and the resulting value of T, for small α, is $T \simeq 2+\sqrt{\alpha}$. Substituting this in (4.18), we see that

$$\lambda_{max} \simeq 1 - \sqrt{2\alpha} \tag{4.20}$$

For small α, then, the FCFS splitting algorithm has the

same maximum throughput as slotted Aloha. This is not surprising, since without CSMA, the major advantage of the FCFS algorithm is its efficiency in resolving collisions, and with CSMA, collisions rarely occur. It is somewhat surprising at first that if we use the FCFS algorithm with equal subintervals (ie. $x=1/2$), then we are limited to a throughput of $1-\frac{1}{3}\alpha$. This degradation is due to a substantial increase in the number of collisions.

The same type of analysis as used here can be used for reservation multiaccess systems and a variety of other conditions. The idea, originally due to Humblet [53] is to generalize our original assumptions a) to e) to allow the durations of idle, success, or collision slots to all be different. Recall that in CSMA, idle slots had duration α and success and collision slots had duration $1+\alpha$. In a reservation system, idle and collision slots would have the duration required to send a reservation packet, whereas success slots would have the duration required for both a reservation and a message transmission.

REFERENCES

1) Abramson, N., "The Aloha System--Another Alternative for Computer Communications", Fall Joint Computer Conference, AFIPS Conf. , Vol. 37, 1970

2) Ahlswede, R., "Multi-way Communication Channels, Proc. 2nd Int. Symp. Inform. Th., Tsahkadsor, Armenian S.S.R., 1971; Hungarian Acad. Sc., pp. 23-52, 1973.

3) Liao, H., "A Coding Theorem for Multiple Access Communications", Int. Symp. Information Th., Asilomar, 1972; also "Multiple Access Channels", Ph.D. Thesis, Dept. of E.E., Univ. Hawaii, 1972.

4) El Gamal, A. and T. M. Cover, Multiple User Information Theory", Proc. of the IEEE, Vol. 68, pp. 1466-1483, Dec. 1980.

5) van der Meulen, E. C., "A Survey of Multi-way Channels in Information Theory, 1961-1976", IEEE Trans. I.T., Vol. IT-23, Jan. 1977.

6. Csiszar, I. and J. Körner, Information Theory: Coding Theorems for Discrete Memoryless Systems, Chap. 3, Academic Press, 1781

7) Special Issue on Spread Sprectrum, IEEE Transactions on Communications, May 1982.

8) Pursley, M. B., "Frequency-Hop Transmission for Satellite Packet Switching and Terrestrial Packet Radio Networks", Report T-144, Coordinated Science Laboratory, Univ. of Ill.

9) Slepian, D., and J. K. Wolf, "A Coding Theorem for Multiple Access Channels with Correlated Sources", Bell System Technical Journal, Vol. 52, pp. 1037-1076, Sept. 1973.

10) Gallager, R. G., Information Theory and Reliable Communication, John Wiley, 1968.

11) Bierbaum, M. and H. M. Wallmeier, "A Note on the Capacity Region of the Multi-Access Channel", IEEE Trans. on IT, July 1979, Vol. IT-25, p. 484.

12) Arimoto, S., "An Algorithm for Computing the Capacity of Arbitrary Discrete Memoryless Channels", IEEE Trans. on Information Theory, Vol. IT-18, No. 1, Jan 1972, pp. 14-20.

13) Blahut, R. E., "Computation of Channel Capacity and Rate-Distortion Function", IEEE Trans. on Information Theory, Vol. IT-18, No. 4, July 1972, pp. 460-470.

14) Massey, J. L. and P. Mathys, "The Collison Channel without Feedback", IEEE Trans. IT, this issue.

15) Arutyunyan, E. A., "Lower Bound for the Error Probability of Multiple-Access Channels", Problemy Peridachi Informatsii, Vol. 11, No. 2, pp. 23-36, April 1975.

16) Peterson, R. L. and D. J. Costello, "Error Probability and Free Distance Bounds for Two-User Tree Codes on Multiple Access Channels", IEEE Trans. IT, Vol. IT-26, pp. 658-670, Nov. 1980.

17) Gaarder, N. T. and J. K. Wolf, "The Capacity Region of a Multiple Access Discrete Memoryless Channel Can Increase with Feedback", IEEE Trans. IT, Vol. IT-21, pp. 100-102, Jan. 1975.

18) Ozarow, L. H., "The Capacity of the White Gaussian Multiple Access Channel With Feedback", IT Trans. on IT, Vol. IT-30, pp. 623-629, July 1984.

19) Cover, T. M., R. J. McEliece, and E. C. Posner, "Asynchronous Multiple-Access Channel Capacity", IEEE Trans. IT, Vol. IT-27, pp. 409-413, July 1981.

20) Hui, J.Y.N. and P. A. Humblet, "The Capacity Region of the Totally Asynchronous Multiple Access Channel", IEEE Trans. IT, this issue.

21) Cohen, A. R., J. A. Heller, and A. J. Viterbi, "A New Coding Technique for Asynchronous Multiple Access Communication", IEEE Trans. Comm. Tech., Vol. COM-19, pp. 849-855, Oct. 1971.

22) Sommer, R. C., "High Efficiency Multiple Access Communication Through a Signal Processing Repeater", IEEE Trans. Comm. Tech., Vol. COM-16, pp. 222-232, April 1968.

23) Narayan, P. and D. L. Snyder, "The Two-User Cutoff Rate for an Asynchronous and a Synchronous Multi-Access Channel Are The Same", IEEE Trans. IT, Vol. IT-27, pp. 414-419, July 1981.

24) Arikan, E., Private Communication.

25) Massey, J., "Capacity, Cutoff Rate, and Coding for a Direct Detection Optical Channel", IEEE Trans. Comm., Vol. COM-29, pp. 1615-1621, Nov. 1981.

26) Humblet, P., "Error Exponents for a Direct Detection Optical Channel", Report LIDS-P-1337, Laboratory for Information and Decision Systems, M.I.T., Oct. 1983.

27) Roberts, L. G., "Aloha Packet System With and Without Slots and Capture. ASS Note 8, ARPA Network Info. Ctr., S.R.I., Stanford, Calif., June 1972.

28) Kaplan, M., "A Sufficient Condition for Non-ergodicity of a Markov Chain", IEEE Trans. IT, Vol. IT-25, pp. 470-471, July 1979.

29) Hajek, B. and T. van Loon, "Decentralized Dynamic Control of a Multiaccess Broadcast Channel", IEEE Trans. Automatic Control, Vol. AC-27, pp. 559-569, June 1982.

30) Ferguson, M. J., "A Study of Unslotted Aloha with Arbitrary Message Lengths", Proc. 4th Data Comm. Symposium, Quebec, Canada, pp. 5.20, 5.25, Oct. 1975.

31) Sant, D., "Throughput of Unslotted Aloha Channels with Arbitrary Packet Interarrival Time Distribution", Proc. IEEE Trans. Comm., Vol. COM-28, pp. 1422-1425, Aug. 1980.

32) Capetanakis, J. I., "The Multiple Access Broadcast Channel: Protocol and Capacity Considerations", Ph.D. Thesis, Dept. of EE & CS, MIT, Aug. 1977; also Abstracts of Int. Symp. on IT, Oct. 1977, Cornell Univ. and IT Trans. IT, Vol. IT-25, pp. 505-515, Sept. 1979.

33) Hayes, J., "An Adaptive Technique for Local Distribution", Bell Telephone Laboratory Technical Memorandum TM-76-3116-1, March 1976; also IEEE Trans. Comm., Vol. COM-26, pp. 1178-1186, Aug. 1978.

34) Tsybakov, B. S. and V. A. Mikhailov, "Free Synchronous Packet Access in a Broadcast Channel with Feedback", Problemy Peredachi Informatsii, Vol. 14, No. 4, pp. 32-59, Oct. 1978.

35) Massey, J. L., Private Communication, 1977. Also "Collision-Resolution Algorithms and Random Access Communications", UCLA Report UCLA-ENG-8016, April 1980.

36) Gallager, R. G., "Conflict Resolution in Random Access Broadcast Networks", Proceedings AFOSR Workshop in Communication Theory and Applications, pp. 74-76, Provincetown, Mass., Sept. 1978.

37) Tsybakov, B. S. and V. A. Mikhailov, "Random Multiple Access of Packets. Part and Try Algorithm", Problemy Peridachi Informatsii, Vol. 16,N4, pp. 65-79, Oct. 1980.

38) Tsybakov, B. S. and N. B. Likhanov, "An Upper Bound on Packet Delay in a Multiple Access Channel", Problemy Periodachi Informatsii, Vol. 18, N4, pp. 76-84, Oct. 1980.

39) Huang, J.-C. and T. Berger, "Delay Analysis of the Modified 0.487 Contention Resolution Algorithm", Cornell Univ.

Report, submitted to IEEE Trans. on Comm.

40) Mosely, J. and P. Humblet, "A Class of Efficient Contention Resolution Algorithms for Multiple Access Channels" to appear in IEEE Trans. on Comm.; also Report LIDS-P-1194, Laboratory for Information and Decision Systems, M.I.T., March 1982; Report LIDS-TH-918, Laboratory for Information and Decision Systems, May 1979.

41) Vvedenskaya, N. D. and M. S. Pinsker, "Non-optimality of the Part-and-Try Algorithm", International Workshop on Convolutional Codes; Multiuser Communication, Abstracts of Papers, Sochi, pp. 141-148, 1983.

42) Pippenger, N., "Bounds on the Performance of Protocols for a Multiple Access Broadcast Channel", IEEE Trans. IT, Vol. IT-27, March 1981.

43) Humblet, P. A., "Bounds on the Utilization of Aloha-like Multiple Access Broadcast Channels", Report LIDS-P-1000, Laboratory for Information and Decision Systems, M.I.T., June 1980.

44) Molle, M. L., "On the Capacity of Infinite Population Multiple Access Protocols", IEEE Trans. IT, Vol. IT-28, pp. 396-401, May 1982 (submitted May 1980).

45) Cruz, R. and B. Hajek, "A New Upper Bound to the Throughput of a Multi-Access Broadcast Channel", IEEE Trans. IT, Vol. IT-28, pp. 402-405, May 1982.

46) Mikhailov, V. A. and B. S. Tsybakov, "Upper Bound for the Capacity of a Random Multiple Access System", Problemy Peridachi Informatsii, Vol. 17, No. 1, pp. 90-95, Jan. 1981.

47) Cruz, R. L., "Protocols for Multiaccess Channels with Continuous Entry and Noisy Feedback", Laboratory for Information and Decision Systems, Report LIDS-TH-1213, May 1982.

48) Tsybakov, B. S., "Survey of USSR Contributions to Random Access Communications", IEEE Trans. on IT, this issue.

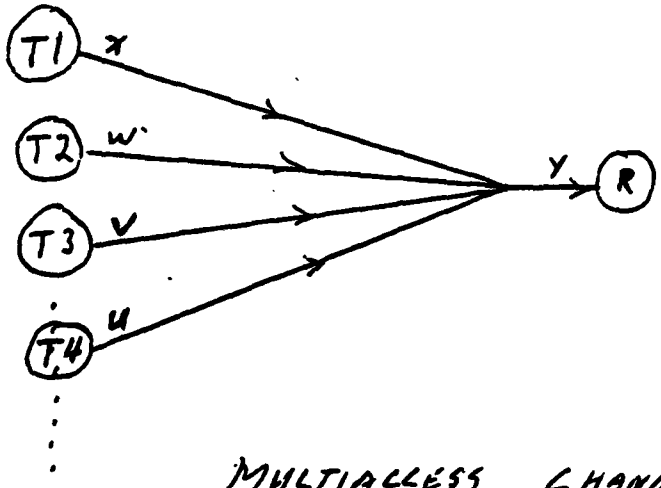
49) Mathys, P. and P. Flajolet, "Q-ary Collision Resolution Algorithms in Random-Access Systems with Free or Blocked Channel Access", Presented at IEEE Int. Symp. on IT, St. Jovite, Quebec, Canada, Sept. 1983; Inst. of Telecommunications, ETH, Zurich.

50) Hluchyj, M. G. and R. G. Gallager, "Multiaccess of a Slotted Channel by Finitely Many Users", National Telecommunications Conference, New Orleans, Dec. 1981; also Laboratory for Information and Decision Systems, Report LIDS-P-1131, Aug. 1981.

51) Berger, T., N. Mehravari, D. Towsley, and J. Wolf, "Random Multiple Access Communications and Group Testing", IEEE Trans. of Comm., Vol. COM-32, pp. 769-779, July 1984.

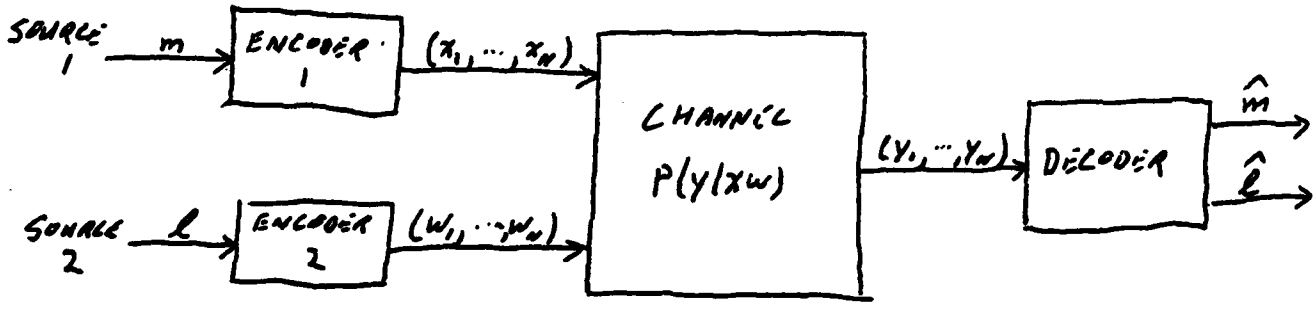
52) Kleinrock, L. and F. A. Tobagi, "Packet Switching in Radio Channels: Part 1: CSMA Modes and Their Throughput-Delay Characteristics", IEEE Trans. on Comm., Vol. COM-23, pp. 1400-1416, Dec. 1975.

53) Humblet, P. A. and J. Mosely, "Efficient Accessing of a Multiaccess Channel", Proceedings of 19th Conference on Decision and Control, Dec. 10-12, 1980, Albuquerque, NM; also in LIDS-P-1040, M.I.T., Cambridge, MA.



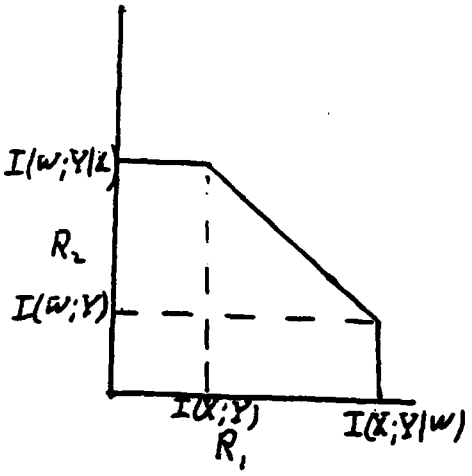
MULTIACCESS CHANNEL

FIGURE 1.1

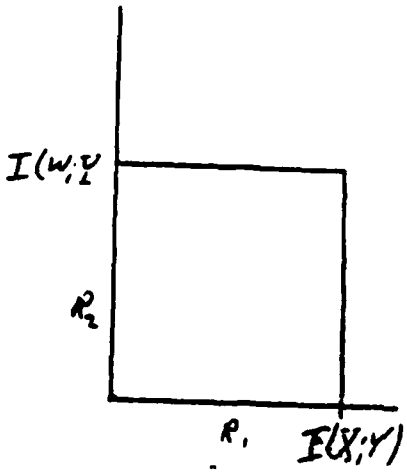


MULTIACCESS CHANNEL WITH TWO TRANSMITTERS

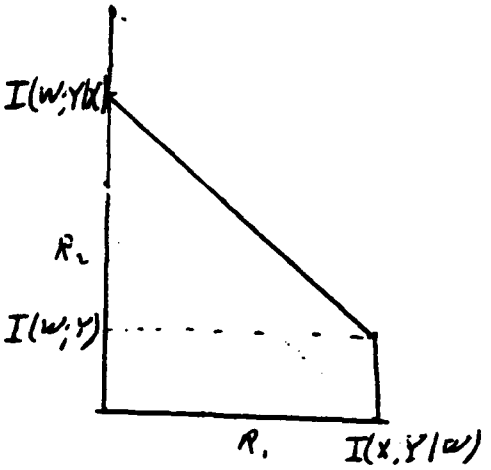
FIGURE 2.1



NORMAL CASE



SPECIAL CASE
$I(X;Y) = I(X;Y|W)$



SPECIAL CASE
$I(X;Y) = 0$

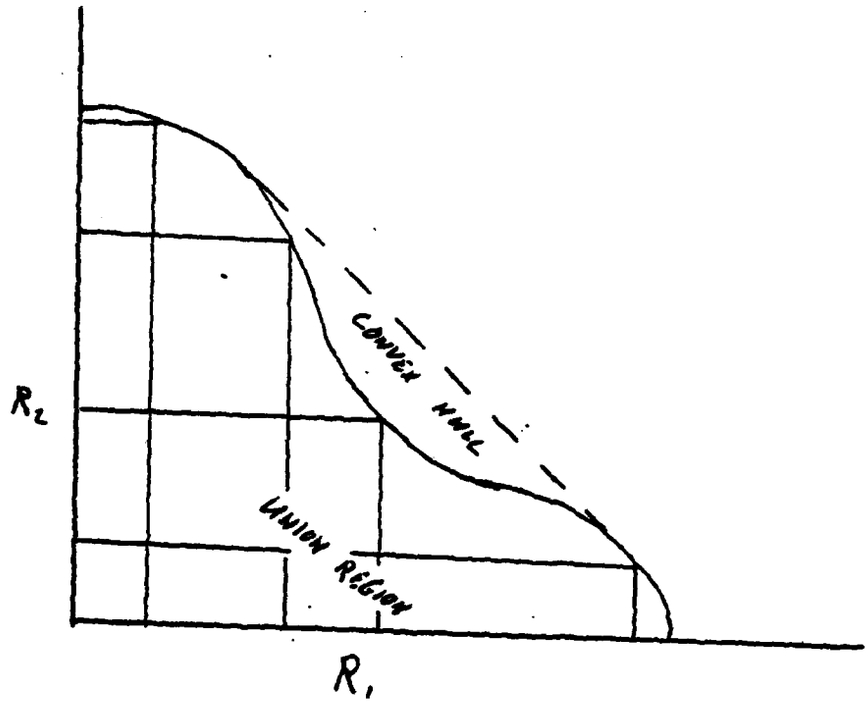
RATE REGION OF EQS. (2.3) - (2.5)

FIG. 2.2

		X = 0	X = 1	X = 2
W	0	(0,0)	(1,0)	(2,0)
	1	(1,0)	(C,C)	(C,C)
	2	(2,0)	(C,C)	(C,C)

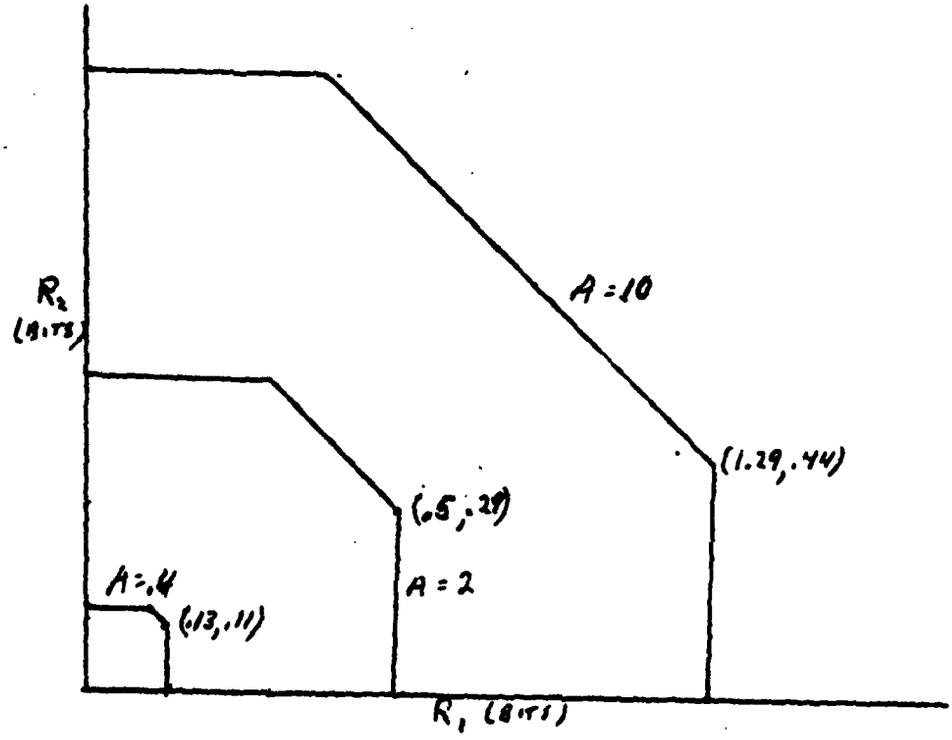
OUTPUT $Y = (Y', Y'')$ AS A FUNCTION OF X, W FOR COLLISION CHANNEL WITH $K = 2$

FIGURE 2.3



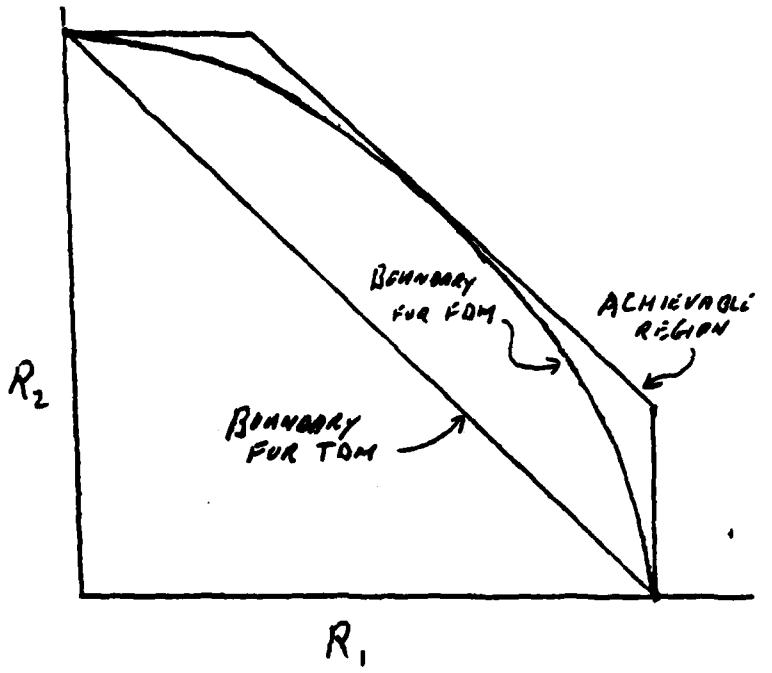
ACHIEVABLE RATE REGION
FOR COLLISION CHANNEL

FIGURE 2.4



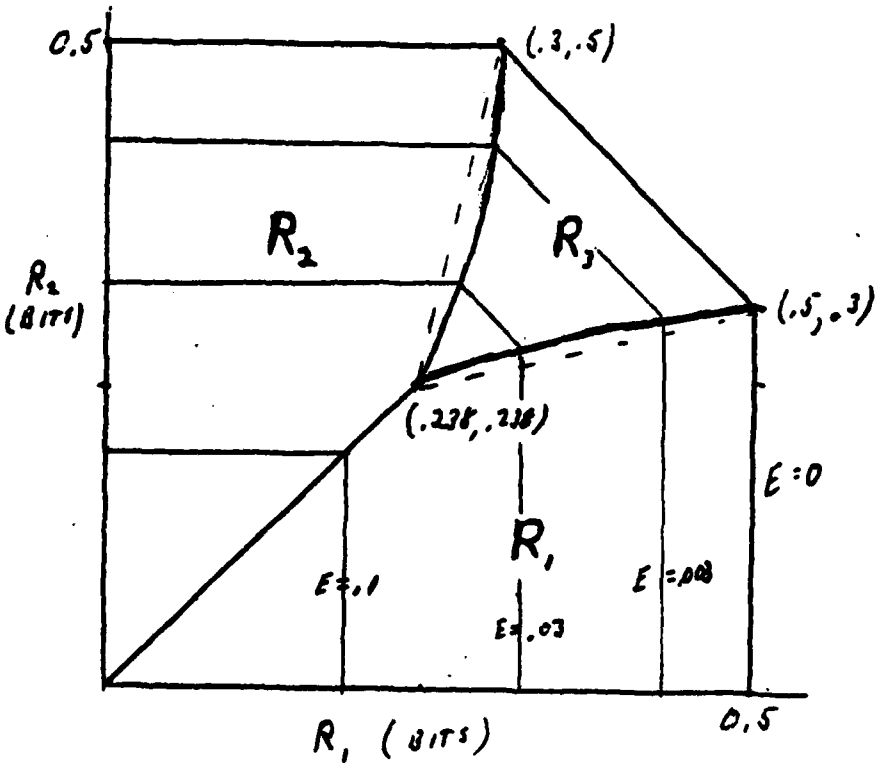
ACHIEVABLE RATE REGION AS FUNCTION OF SIGNAL/NOISE A

FIGURE 2.5



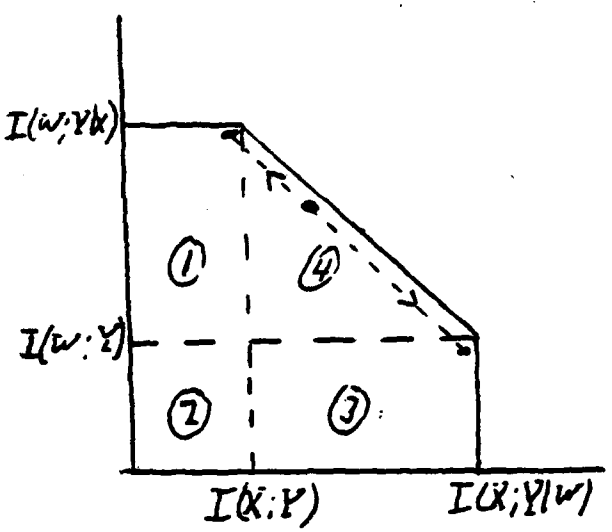
COMPARISON OF TDM (AS MODELLED) AND FDM

FIGURE 2.6



REGIONS R_i WHERE $E_{r,i}$ DOMINATES ERROR BOUND; FOR A=1.

FIGURE 2.7



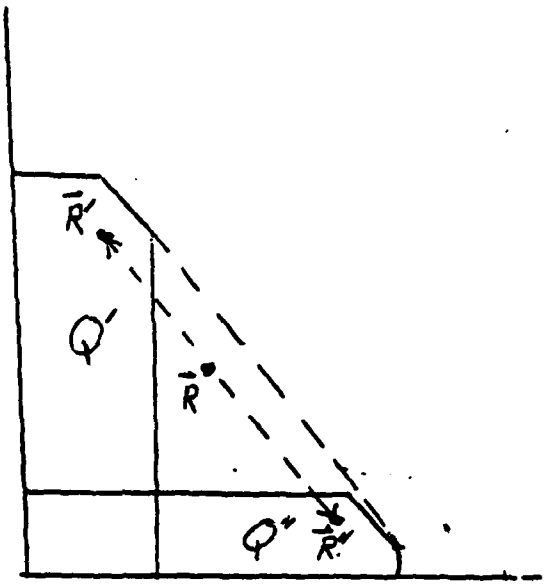
REGION 1: DECODE X, THEN W GIVEN X

REGION 2: DECODE X, W INDEPENDENTLY

REGION 3: DECODE W, THEN X GIVEN W.

REGION 4: USE TIME SHARING BETWEEN 1, 3.

a)

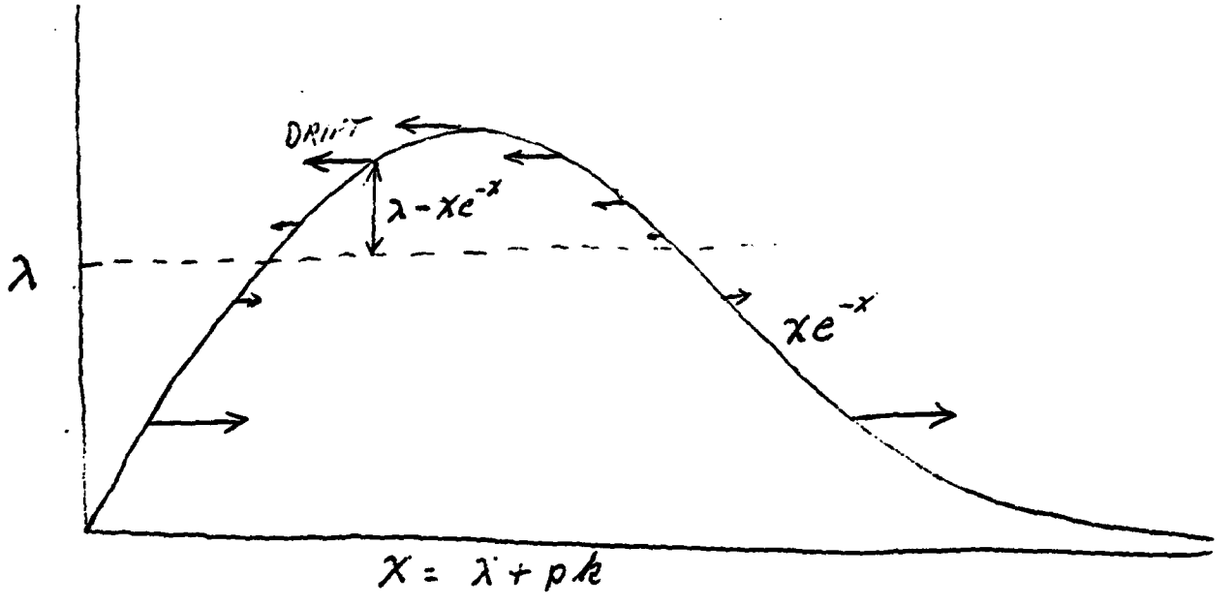


TIMESHARING OF POINT $\vec{R}$ IN CONVEX HULL BETWEEN $\vec{R}'$ AND $\vec{R}''$

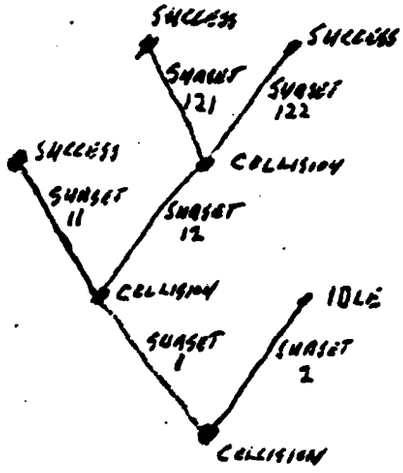
b)

TIMESHARING CONSTRUCTION TO OBTAIN ANY ACHIEVABLE RATE PAIR WITHOUT JOINT DECODING

FIGURE 3.1



D_k AS A FUNCTION OF λ, p, k
FIGURE 4.1

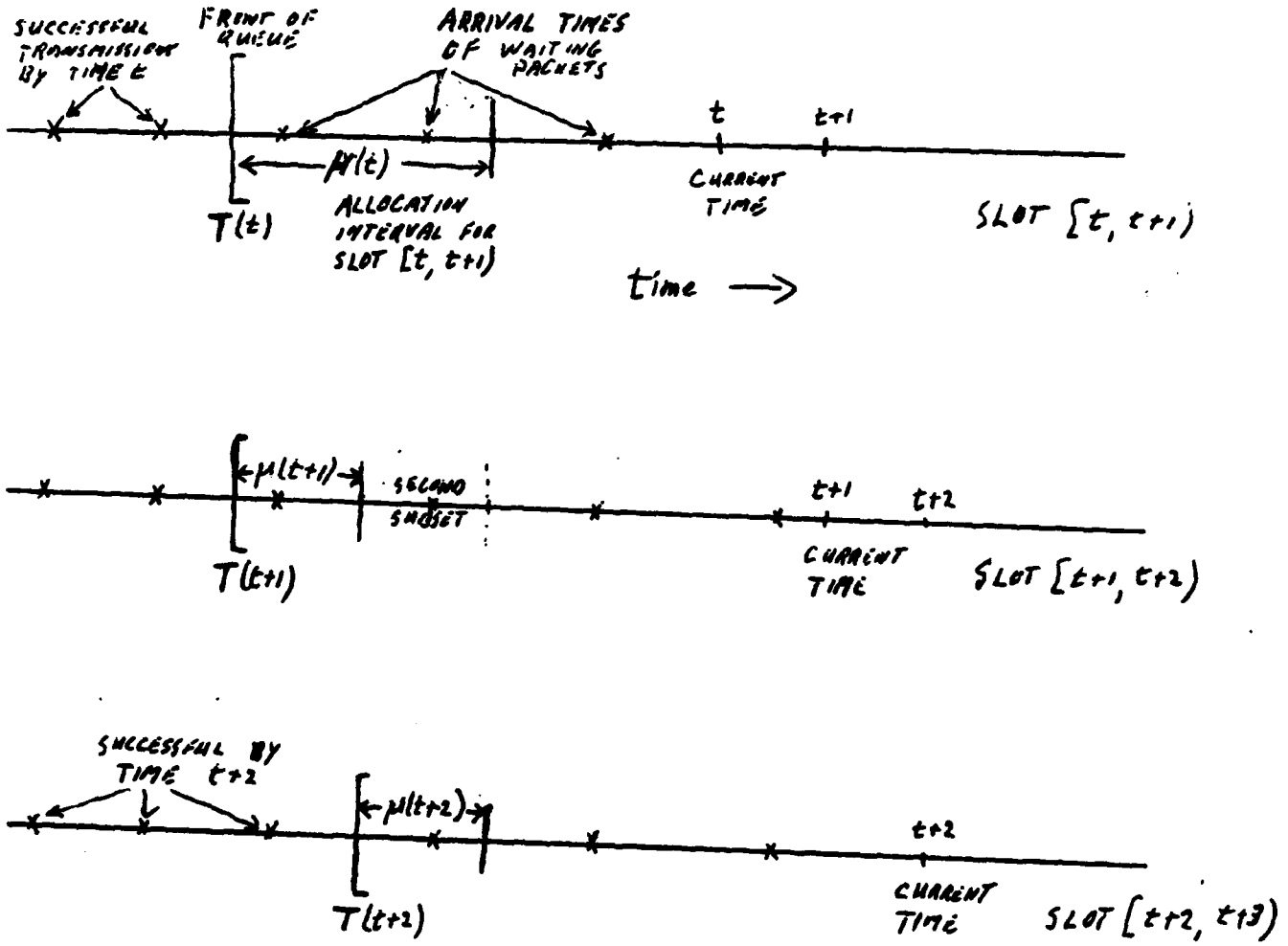


ORDER OF TRANSMISSION AFTER INITIAL COLLISION

1) SUBSET 1
2) SUBSET 11 (SUBSET OF SUBSET 1)
3) SUBSET 12 (OTHER SUBSET OF SUBSET 1)
4) SUBSET 121
5) SUBSET 122
6) SUBSET 2

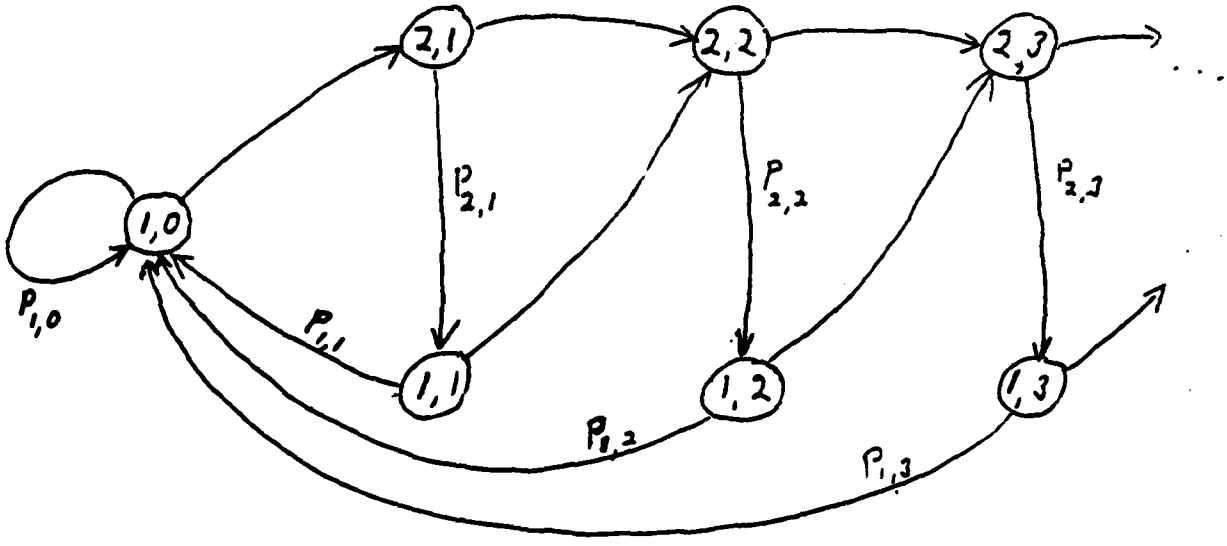
TREE ALGORITHM FOR COLLI

FIGURE 4.2



RECORD OF A COLLISION RESOLUTION PERIOD FOR FCFS SPLITTING ALGORITHM

FIGURE 4.3



MARKOV CHAIN FOR COLLISION RESOLUTION PERIOD

FIGURE 4.4